NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

DESIGNING A MACHINERY CONTROL SYSTEM (MCS) SECURITY TESTBED

by

Nathan H. Desso

September 2014

Thesis Advisor:	Thuy D. Nguyen
Thesis Co-Advisor:	Mark Gondree

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2014	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE DESIGNING A MACHINERY CONTROL SYSTEM (MCS) SECURITY TESTBED	**5. FUNDING NUMBERS**
6. AUTHOR(S) Nathan H. Desso	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. government. IRB protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Industrial control systems (ICS) face daily cyber security threats, can have a significant impact to the security of our nation, and present a difficult challenge to defend. Critical infrastructures, including military systems like the machinery control systems (MCS) found onboard modern U.S. warships, are affected because of their use of commercial automation solutions. The increase of automated control systems within the U.S. Navy saves in manpower costs but increases the need for cyber security research and defense. Research is needed to assess and contribute solutions to ICS security problems.

This thesis describes the MCS security testbed, which supports research in the security of shipboard machinery control systems. The testbed has been conceptualized, designed and implemented with the vision of supporting research and experimentation on the defense of ICS and MCS systems. The testbed provides the ability to analyze vulnerabilities, test defenses and replicate attacks on authentic physical industrial control equipment. The MCS security testbed is a tool that may help counter cyber security threats facing the defense industrial base today. Future solutions to attacks on control systems in the nation's critical infrastructure begin with experimentation using authentic test environments.

14. SUBJECT TERMS cyber security, critical infrastructure, supervisory control and data acquisition, industrial control system, machinery control system, programmable logic controller, security testbed	**15. NUMBER OF PAGES** 183
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

DESIGNING A MACHINERY CONTROL SYSTEM (MCS) SECURITY TESTBED

Nathan H. Desso
Lieutenant, United States Navy
B.A., The Citadel Military College, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2014

Author: Nathan H. Desso

Approved by: Thuy D. Nguyen
Thesis Advisor

Mark Gondree
Thesis Co-Advisor

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Industrial control systems (ICS) face daily cyber security threats, can have a significant impact to the security of our nation, and present a difficult challenge to defend. Critical infrastructures, including military systems like the machinery control systems (MCS) found onboard modern U.S. warships, are affected because of their use of commercial automation solutions. The increase of automated control systems within the U.S. Navy saves in manpower costs but increases the need for cyber security research and defense. Research is needed to assess and contribute solutions to ICS security problems.

This thesis describes the MCS security testbed, which supports research in the security of shipboard machinery control systems. The testbed has been conceptualized, designed and implemented with the vision of supporting research and experimentation on the defense of ICS and MCS systems. The testbed provides the ability to analyze vulnerabilities, test defenses and replicate attacks on authentic physical industrial control equipment. The MCS security testbed is a tool that may help counter cyber security threats facing the defense industrial base today. Future solutions to attacks on control systems in the nation’s critical infrastructure begin with experimentation using authentic test environments.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AB	Allen Bradley
AFTL	analog fluid tank lab
CCS	central control station
CONOPS	concept of operations
COTS	commercial of the shelf
CPU	central processing unit
CVN	nuclear aircraft carrier
DCS	distributed control system
DDG	guided missile destroyer
DIOL	digital input output lab
FLSD	fluid level simulator dial
FLT	fault
GUI	graphical user interface
HMI	human machine interface
HOS	human machine interface override station
ISCS	integrated ship control system
I/O	input and output
IDE	integrated development environment
IEC	international electrotechnical commission
LAN	local area network
LED	light emitting diode
LHD	landing helicopter dock amphibious ship
LVL	level
MCM	mine countermeasure ship
MCS	machinery control systems
NIC	network interface card
NPS	Naval Postgraduate School
OS	operating system
PLC	programmable logic controller
REM	remote run

SC MCS	smart carrier machinery control system
SCADA	supervisory control and data acquisition
SPST	single pole single throw
TF	functional test
TE	exception test
VDC	volts direct current
VLV	valve
VM	virtual machine

ACKNOWLEDGMENTS

I would like to express thanks to my wife and children for their patience and the sometimes-needed distraction that provided me with the motivation I so desperately needed through this entire thesis process.

I would also like to thank my parents, Hoyt and Maryann Desso, for always telling me that I can do anything that I put my mind to. The encouragement, structure and basic core values you gave me have always guided me to success.

I would like to thank Prof. Mark Gondree and Prof. Thuy Nguyen as my advisory team. Thank you for your patience, assistance and dedication to seeing me through this process.

Finally, I would like to thank God for his hand in molding me into the person that I am today. I give thanks to Him for the many blessings that He has bestowed upon me.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

On a daily basis, the cyber security operations community identifies vulnerabilities in Supervisory Control and Data Acquisition (SCADA) systems, industrial control systems (ICS's) and machinery control systems (MCS's). Today's modern U.S. warships use control systems that are similar to, or the same as, those found in the civilian industry. Critical industrial control systems like those found in electrical networks, MCS's onboard warships, water storage facilities and oil refineries constantly require patches or monitoring for newly discovered security flaws. The concern over vulnerabilities in SCADA systems is due to the equipment they control and their impact, as an unprotected security flaw can lead to loss of a critical capability, loss of life or limb, and a threat to national security.

According to DHS's 2013 ICS-CERT year-in-review, there was an average of one ICS incident every 34 hours, for a total of 257 incidents reported in the United States [1]. In July 2014, ICS customers seeking a PLC software update inadvertently downloaded a maliciously planted Trojan, dubbed Havex. The Havex Trojan scans a widely used PLC standard called open protocol communications (OPC) and is believed to gather information about vulnerabilities in the target system, as a way to test code before a larger attack. The Havex Trojan affected a German manufacturing company, a Belgium PLC VPN software company and a Swiss company that produces industrial cameras [2], [3]. Another cyber attack called "Careto" or "The Mask" was a widespread espionage malware attack targeted at government agencies, energy, oil and gas companies and research organizations. The malware intercepts all communication channels and collects vital information from the victim's machine, to facilitate the theft of private information from ICS's [4]. Relatedly, recent vulnerabilities discovered amongst maritime ships using global positioning systems (GPS), automatic identification systems (AIS), and electronic chart display and information system (ECDIS) have uncovered serious flaws in security [5]. Flaws in maritime systems give hackers the ability to change the direction of the ship using false GPS data, manipulate chart data and shut down the ships ability to communicate with ports or other ships using AIS [5]. The U.S. Navy uses all of these systems and is exploring ways to integrate ICS's with navigation systems like ECDIS and AIS in a push toward autonomous warships [6]. In the U.S. Department of Defense (DOD) the criticality of ICS and MCS systems are directly linked to the defense of the nation, and in the case of the Navy, directly linked to warships and national assets.

Learning the vulnerabilities of the ICS and MCS systems and having the ability to test for possible defenses using an authentic testbed is valuable to cyber security. In particular, this line of research is in direct support of *Presidential Policy Directive PPD-21* and *Presidential Decision Directive 63* to secure critical infrastructure from cyber attack [7], [8]. The ability to test using actual industry standard equipment in real time will provide valuable insight into identifying vulnerabilities applicable to both the military and civilian domains. A testbed has been used in other colleges and universities to educate researchers on SCADA [9], [10], [11], but limited work has been done targeting U.S. Navy or military applications. The testbed described in this thesis is built to assess system communication vulnerabilities and allow for simulated attacks on authentic physical systems without damaging expensive equipment or placing personnel at risk. The testbed provides a testing environment for both control and monitoring of machinery systems similar to the military and civilian industry.

B. THESIS OVERVIEW

In Chapter I, we provide motivation for the MCS security testbed. In Chapter II, we discuss the background of ICS and MCS in the U.S. Navy and the major components that make up an MCS. In Chapter III, we discuss the design of the MCS security testbed. In Chapter IV, we describe the analog fluid tank lab (AFTL) concept of operations (CONOPS), design, implementation and functional requirements. In Chapter V, we describe the digital I/O lab (DIOL) CONOPS, design, implementation and functional requirements. In Chapter VI, we discuss the testing plan designed to support the testbed requirements. In Chapter VII, we conclude.

II. BACKGROUND

Control systems in general are collections of hardware, software, communication networks (or media) that are used to monitor and control processes, services, or commodities [12]. There are two primary types of control systems. Distributed Control Systems (DCS) typically are used within a single processing or generating plant or over a small geographic area. Supervisory Control and Data Acquisition (SCADA) systems typically are used for large, geographically dispersed distribution operations. For example a utility company may use a DCS to generate power and a SCADA system (or network) to distribute the control and monitoring of it [13]. In the U.S. Navy a machinery control system (MCS) is a type of DCS used to remotely actuate, control and monitor machinery equipment like engines, electrical plants and auxiliary systems [14]. The U.S. Navy MCS combines many systems into one network to both operate and monitor the entire engineering plant utilizing different communication protocols but does not span a large geographical area. Because of the small geographical area covered, by definition the MCS's of the U.S. Navy are not SCADA systems. However, most recently amongst the cyber security and naval systems engineering personnel the term SCADA has become synonymous with both DCS and MCS type systems. Since the naval MCS systems use the exact commercial off the shelf (COTS) components as found in SCADA systems, the term SCADA MCS will be used in the remainder of this thesis to mean "the network of digital and analog commercial components found aboard naval ships that are monitoring and controlling naval machinery systems" [6], [14].

Machinery control systems have evolved significantly in both the military and civilian applications from the direct wire and gauge days of the past to automated digital controls and displays of today. Machinery controls have moved from hardware-based logic to software-based logic where lights and pushbuttons are replaced by processors, graphical user interfaces and keyboards [14]. Despite the advantages, the U.S. Navy initially showed reluctance to move from hardware centric systems to software centric systems for fear of relinquishing control to automated systems. However, the cost savings in personnel associated with adopting a software centric COTS system proved to be enough to move forward with the modernization and upgrade of current and future shipboard MCS systems [14]. The U.S Navy currently utilizes software PLC centric machinery control systems to operate all variations of systems on multiple ships from mine countermeasure ships to aircraft carriers. Historically, on older systems the major components include control consoles or panels, hard-wired analog interfaces to sensors and equipment, and a network of hardware to remotely control each major component in the engineering system. Today the systems have evolved to the digital and analog computer centric networked systems found on modern aircraft carriers (CVN), guided missile destroyers (DDG) and large amphibious platforms like the USS *Makin Island* (LHD-8). It is these systems that have reduced manpower requirements, increased information flow and automated many engineering processes that present challenges in security and require a navy testbed for experimentation [14].

MCS systems of today are complex networks that are comprised of engineering workstations, human machine interfaces (HMI), PLC's and a connected network centered on a fiber optic or Ethernet backbone (see Figure 1). The engineering workstation is a computer workstation (usually Windows based) that a technician can use to load software, monitor the PLC's health and adjust or configure the support network. PLC's are processor and I/O units connected to the MCS network that operate on logic software and directly monitor and control equipment. The PLC racks are comprised of central processing units (CPU) and input output (I/O) modules that share information via a common internal bus as well as interface with the machinery plant directly. HMI's are graphical user interfaces that allow the operator to monitor and control systems remotely through the network. The network that connects these components utilizes Ethernet or fiber optic cables interconnecting hubs that will direct and route information and controls for various systems.

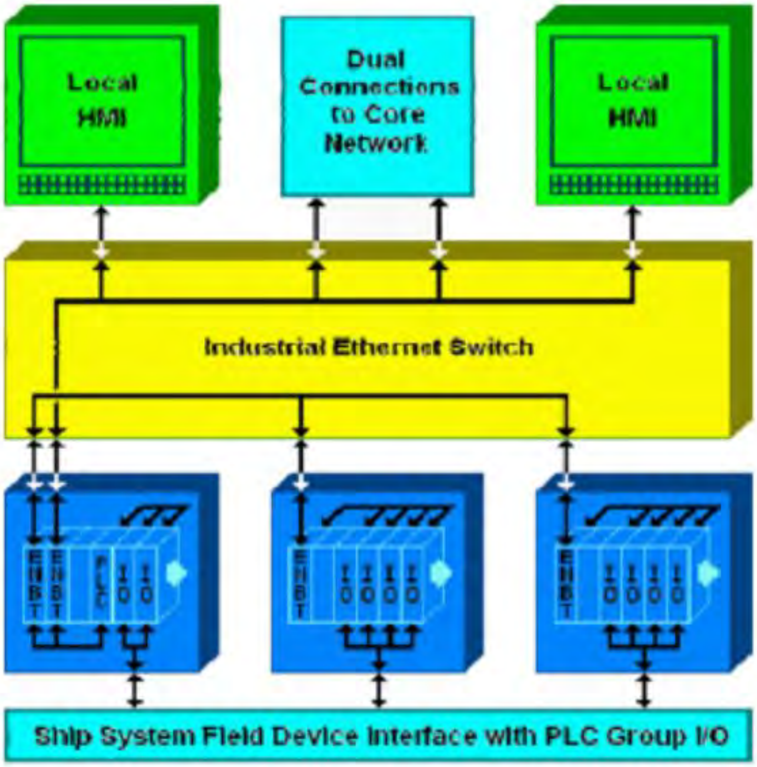
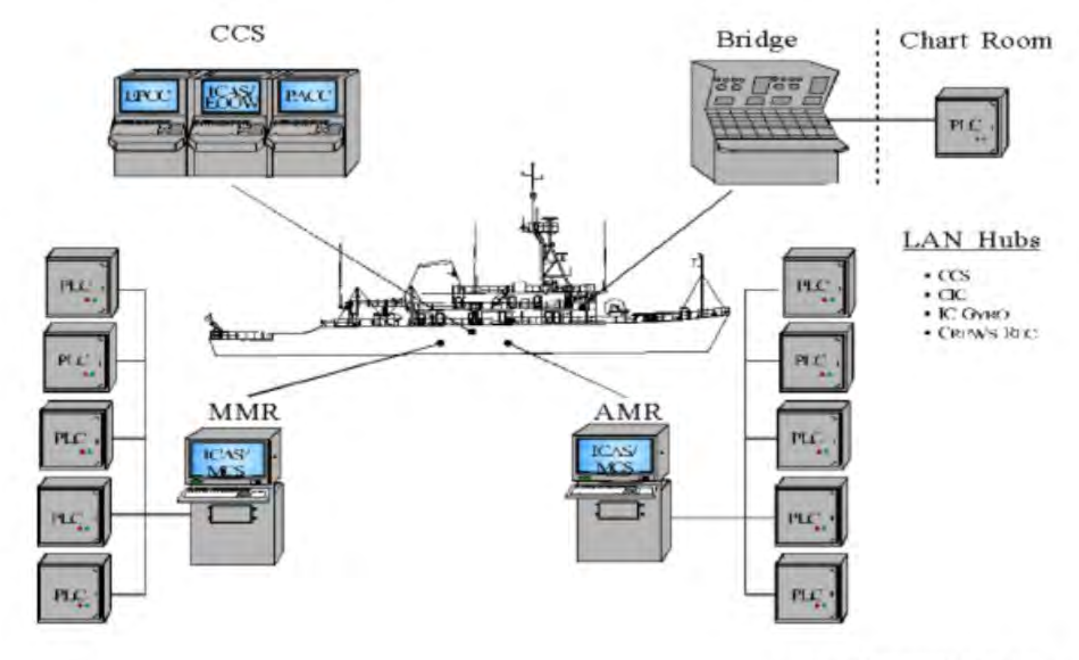

Local HMI
Dual Connections to Core Network
Local HMI
Industrial Ethernet Switch
ENBT
ENBT
PLC
IO
IO
ENBT
IO
IO
IO
IO
ENBT
IO
IO
IO
IO
Ship System Field Device Interface with PLC Group I/O

Figure 1. Basic PLC Group used in MCS systems, from [14]

In the U.S. Navy, the progression from the old hardware systems to the current software based systems began with the DDG-51 class ships and continues through the future CVN-78 systems. The DDG-51 system was the first to be outfitted with all digital MCS systems. The advantages offered by the DDG-51 system over control systems of the past were easy setpoint adjustment and quicker information flow to the operator [14]. The next naval system to improve on MCS design was the MCM-1 system that replaced the Integrated Ship Control System (ISCS) with a COTS system that reduced crew workload, provided state of the art commercial control systems with 24-hour support, provided on-board crew training and interfaced with the original ISCS hardware seamlessly. The MCM-1 class MCS was the first class-wide modern control system [14]. The MCM-1 system consisted of two PLC cabinets (one with a processor with logic and an analog I/O module and the other with a digital I/O module) and three generator local control electronic enclosures. The MCM-1 MCS interconnected all the PLC's using Ethernet connections to local area network (LAN) hubs that exchanged information while using fiber optic connections as a backbone for redundancy. The PLC's in the MCM-1 system were the heart of the system and allowed the operator to monitor many systems but only control one at a time. The signal routing for commands and monitoring was initiated from a console or HMI located in the central control station (CCS) or locally in each engine room. The signal from the HMI would be sent to the PLC and output by the PLC would control the equipment as desired. The feedback as to the status of the monitored equipment was received by the PLC and displayed back on the HMI (see Figure 2). The advantages of the system were readily apparent and ready to be adapted to larger platforms like the CVN [14].



NAVAL ENGINEERS JOURNAL

Figure 2. MCM-1 digital MCS system, from [14]

The U.S. aircraft carriers (CVN), before 2000, operated all auxiliary systems by hand or using hard-wired remote control and alarm panels operate them. The control systems in use were very labor intense and did not provide much in the way of situational awareness to the operator. The smart carrier machinery control system (SC MCS) was developed to incorporate the network of controls and information from the currently installed systems into a digital network that can be monitored and controlled remotely [14]. The SC MCS combined systems like fuel, fire main pumps, potable water, bilge and drain (to name a few) into one state of the art MCS system similar to that of the MCM-1 system. However, unlike the MCM-1 system the SC MCS system was the first MCS to use a complete Ethernet LAN (no fiber optic was used) [14]. The system comprised of as that of the MCM-1 system; it had HMI's, PLC's with interface directly to machinery and an Ethernet hub backbone. The SC MCS system was innovative for the CVN ships because it works with existing systems and requires less manpower while providing better equipment status. The future CVN's (starting with CVN-78) will be built from the very beginning with the SC MCS systems in an effort to further incorporate ships systems together [14].

The MCS systems in the U.S. Navy have allowed the improvement of quality information to be readily available while reducing the number of operators and technicians needed. However, the increasing size and capabilities of these MCS networks and their integration into other systems in the future (like the littoral combat ship or LCS) make them increasingly vulnerable to malicious software, hardware or other network attack vectors yet to be discovered [14]. The Navy cyber security professionals are aware of the threats and vulnerabilities facing the SCADA systems used and are constantly working to improve security of these systems. However, there is always a need to perform cyber security testing on MCS systems without impacting real equipment operation or the deployment schedules of national assets like aircraft carriers.

In the civilian sector SCADA network testbeds have been designed and implemented to test threats to various industries and the networks that they utilize with great success. Some systems like that at Mississippi State University (MSU) even utilize working scaled models of the machinery systems they are replicating [10]. The testbeds at MSU use two different communication types between the HMI and PLC and various types of digital and analog control and monitoring I/O signals. The signals used on the MSU testbed are used to simulate storage tanks, water towers, factory conveyors and smart grid transmission systems and provide multiple research points for the cyber security, systems engineering and electrical engineering domains [10]. Other testbeds like those designed by researchers at UC Berkeley, Carnegie Mellon University and Vanderbilt University are completely emulated networks of virtual components where commercial operating software runs on equivalent hardware while receiving simulated sensor data [9]. The emulation of the testbed components and sensor data is flexible, convenient, and inexpensive but the U.S. Navy requires the ability to test specific manufacturers hardware and software in order to target specific vulnerabilities that may exists only in MCS systems contracted by the U.S. Navy. Despite differences among existing testbeds, the use of a testbed to discover and experiment is very valuable to research and testing both in the military and in the civilian industry.

III. TESTBED DESIGN

The NPS MCS security testbed designed in this thesis provides a working model that replicates the MCS systems in the fleet today. The testbed utilizes components normally found in commercial MCS systems allowing for realistic cyber security testing and research without concern of damaging expensive equipment or effecting operational readiness of a national asset like a CVN. The NPS MCS testbed is designed to allow experimentation and future research in the areas of SCADA cyber defense and attack in order to improve the security of current implemented systems and possibly provide input into future MCS designs.

The machinery control system security testbed consists of two test systems: an analog fluid tank lab (AFTL) and a digital I/O lab (DIOL) (see Figure 3). These test systems share a programmable logic controller (PLC) rack, a human machine interface (HMI) system running as a virtual machine, and an engineering workstation used to program the PLC (Table 1). The two test systems allow for future study of vulnerabilities and malicious code in SCADA MCS systems in an offline controlled lab setting. Each test system is described in this chapter.

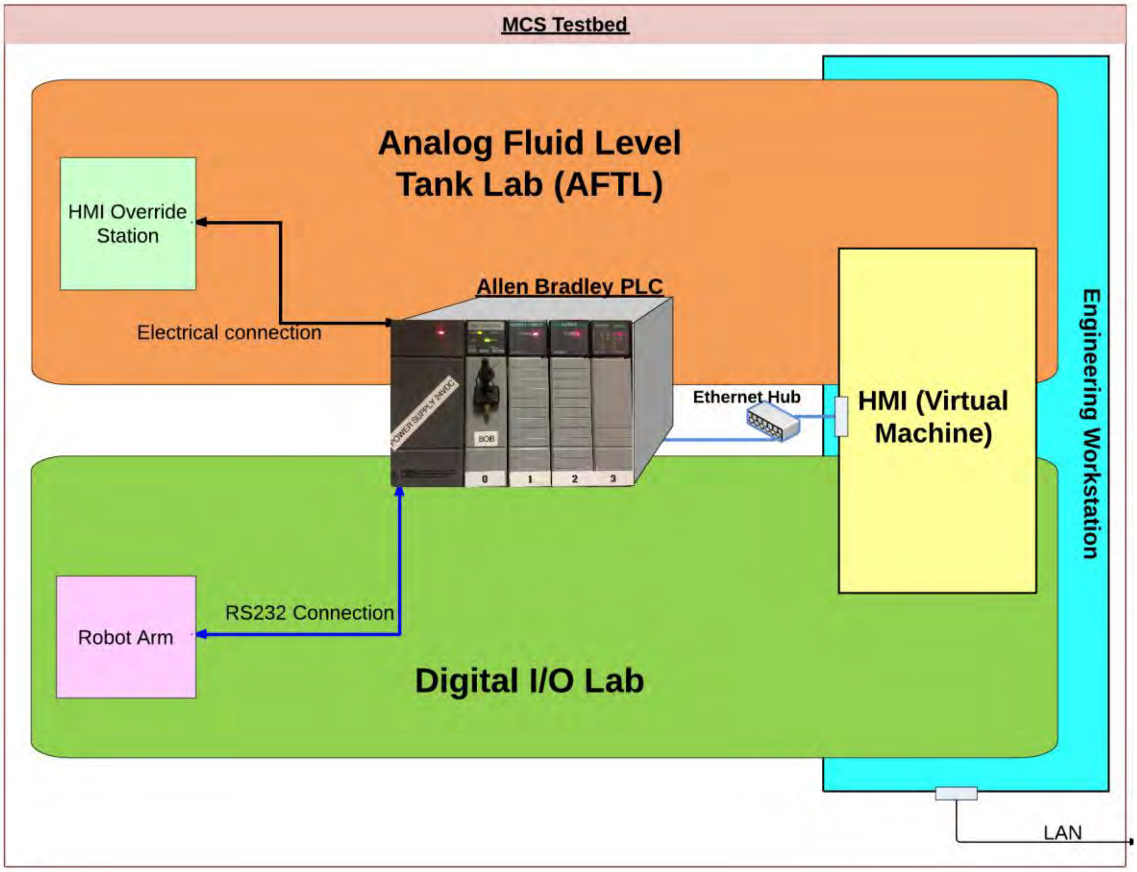


Figure 3. Testbed overview diagram

A. ANALOG FLUID TANK LAB

The analog fluid tank lab models the fluid and storage tank systems found onboard most U.S. Navy ships and in industry. The AFTL uses analog command and sensor signals to control a pump and drain valve in order to maintain the fluid level in a simulated tank. Most control systems support an HMI-override (HOS) mode of operation where a local operator can take control of the PLC in the event of an emergency. The AFTL also provides the HOS capability.

The following functional requirements are levied on the AFTL (Table 1).

Table 1. AFTL functional requirements

ID	Description
R1	The AFTL shall dynamically display the level of the fluid tank to the operator at all stations.
R2	The AFTL shall allow the operator to control the fill pump.
R3	The AFTL shall dynamically display the status of the fill pump.
R4	The AFTL shall allow the operator to control the drain valve.
R5	The AFTL shall dynamically display the status of the drain valve.
R6	The AFTL shall alert the operator when the tank level is above or below its target level.
R7	The AFTL shall allow the operator to adjust the target level parameter during normal operation.
R8	The AFTL shall provide the ability to operate in fully automatic mode, to control the fill pump and drain valve in order to maintain the fluid level at its target level without operator intervention.
R9	The AFTL shall allow the operator to have exclusive and direct system control in the event of an emergency.
R10	The AFTL shall communicate the status of the tank fluid level to the control segment using continuous, analog signals.
R11	The AFTL shall provide the ability to monitor Ethernet traffic between the control and supervisory network segments.

B. DIGITAL I/O LAB

The digital I/O lab demonstrates the data flow between the PLC, HMI, and a digital field device—a robot arm. Proximity sensors will sense the location of the object to be moved. The sensors provide feedback to ensure the proper command is issued to the robot arm to move the object. The data transfer from HMI to robot arm is not simply a signal that changes a bit inside the PLC like the AFTL; the DIOL requires a complete, properly sequenced and formatted digital command signal in order to produce the desired output. The DIOL is designed to allow for testing of the command signals from the HMI to the PLC and the robot arm. The DIOL used in this testbed is a re-implementation of earlier work here at NPS by Ward that focuses more on security of a SCADA network from a protocol and theory standpoint and focuses less on design and implementation of a testbed [15,12].

The following functional requirements are levied on the digital I/O lab (Table 2).

Table 2. Digital I/O lab functional requirements

ID	Description
R12	The DIOL shall operate a robot arm to move objects as commanded.
R13	The DIOL shall display the status of all proximity sensors.
R14	The DIOL shall display the readiness status of the robot arm.
R15	The DIOL shall display all commands sent to the robot arm as they are executed.
R16	The DIOL shall indicate when an error has occurred or exceptional condition exists
R17	The DIOL shall only move objects to proximity sensors that are inactive and shall not move the robot arm when no objects are detected by the proximity sensors.

C. PROGRAMMABLE LOGIC CONTROLLER

The programmable logic controller (PLC) is the heart of an MCS because it is where all sensors and command signals pass through in order to control the field devices. The PLC used in the testbed is a combination of Allen Bradley components. The CPU is an Allen Bradley SLC-5/05 CPU (slot 0) attached to a 1746-A4 4-slot I/O chassis powered by a 1746-P1 onboard power supply with three I/O modules: the AB 1746-NIO4V (slot 1), AB 1746-NO4V (slot 2) and AB 1746-IO12DC (slot 3) (see Figure 4). The PLC CPU is connected to the HMI and Engineering Workstation via Ethernet. Sensors and other analog devices are wired to each I/O module using copper wiring (see Appendices A–B). The sensors, I/O controls, LED indicators and power are wired directly to the 1746-NIO4V, 1746-NO4V and 1746-IO12DC I/O modules. The robot arm is connected as part of the DIOL to the CPU module via an RS-232 interface [16], [17], [18].

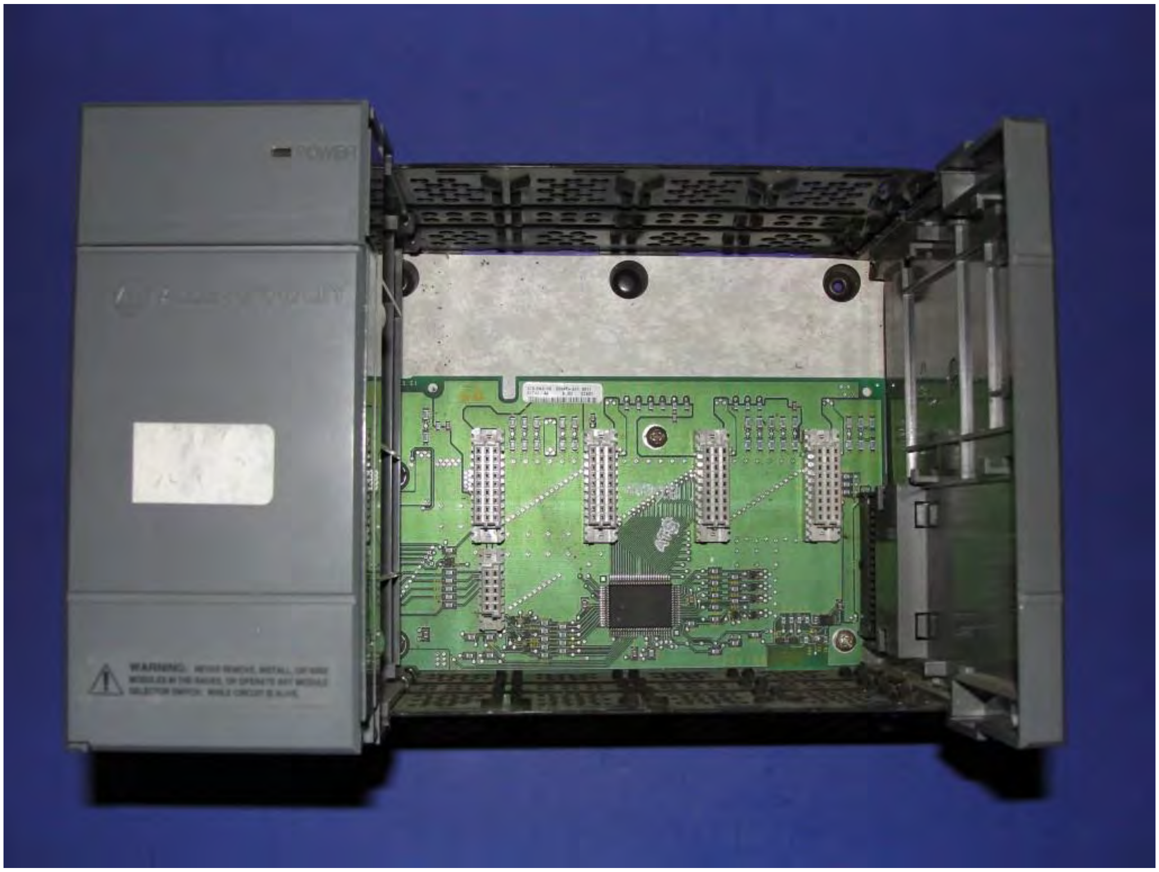


Figure 4. Allen Bradley 1746-A4 (4-slot I/O chassis) with power supply

D. HUMAN MACHINE INTERFACE

The human machine interface (HMI) is a computer that has the ability to remotely control the PLC and monitor its status. The HMI in the MCS testbed interfaces with the PLC using an Ethernet connection that can be monitored by other computers for testing and experiments. Using the HMI, a tester can inject commands to the PLC, monitor how the data is formatted and sent to the PLC, and observing the output.

E. DEVELOPMENT ENVIRONMENT

The development environment is the computer hardware and software that is used to create the ladder logic program running on the PLC, the HMI graphical interface and network access used to communicate with the PLC. The development environment in the MCS testbed consists of a physical host machine and a guest virtual machine for use as the HMI; the host machine acts as an engineering workstation where a technician or engineer can make changes or implement updates to the HMI or PLC. The engineering workstation allows testing and experimentation on ladder logic changes, interception or manipulation of firmware loads and traffic capture.

F. OPERATORS AND THEIR ROLES

The MCS testbed requires three operators in order to function properly. Each operator is present at a station that has a specific function in the MCS testbed. The stations each have independent controls and receive or display some type of information about the MCS testbed status. Each operator is a "human in the loop" who will interpret the data displayed or indicated and issue commands accordingly. The operators needed for the MCS are listed in the following paragraphs.

1. HMI Operator

The HMI operator is located at the HMI and can monitor the graphical display of the AFTL or the DIOL. The HMI operator can observe the status of the field devices via the graphical indications as if they were seen at a remote station. Further, the operator can start, stop or command the analog and digital equipment.

2. HMI-Override Station Operator

The HMI override station (HOS) operator is located at the HOS and can monitor the status of the AFTL and manually controls the fill pump and the drain valve. The HOS operator simulates the operator that would use the HOS to control the equipment directly in the event of an emergency. The HOS uses the input signals to prompt the PLC's ladder logic to act independently of the HMI inputs or any changes that may be influencing the PLC. The HOS operator has as close to manual control of the AFTL as possible.

3. Fluid Level Simulation Dial Operator

The FLS operator is located at the fluid level simulator dial (FLSD) portion of the HOS and can increase or decrease the tank level via a manual dial—the simulation operator can adjust the FLSD, simulating the raising or lowering of the fluid level in the

AFTL “tank.” The level indicated on the dial will be continuously updated and displayed on the HMI. The operator is able to adjust the FLSD for testing or simulation purposes.

IV. ANALOG FLUID TANK LAB

The Analog Fluid Tank Lab (AFTL) is the component of the MCS testbed that provides a platform to experiment with MCS systems using analog I/O. The AFTL is intended to resemble an authentic cyber-physical system that controls field equipment by monitoring continuous, or analog, I/O signals. The AFTL simulates a tank-level monitoring system, a type of system commonly found in shipboard machinery control systems. An Allen Bradley programmable logic controller (PLC) allows experimentation with various controls and outputs communicated over an Ethernet/IP connection. The AFTL offers the ability to analyze and experiment with the network traffic associated with controlling analog field devices in the testbed.

A. CONCEPT OF OPERATIONS

The AFTL attempts to resemble an authentic tank-level control system, which manipulates a fill pump device and drain value to either empty or fill a tank, affecting the fluid tank level. Functional requirements of the AFTL are listed in Table 1. The AFTL is comprised of four primary components: the HMI machine (supervisory segment), the PLC rack (control segment), the HMI override station, and simulated field devices. The system operates in two basic modes of operation: manual and automatic. We describe each of these in more detail, below.

1. AFTL Human Machine Interface

The HMI system presents a graphical interface of the tank level system, allowing the operator to control and monitor the status of the tank level. The HMI displays the following: the status of the fluid tank level, the fill pump, the drain valve; the current mode of operation; alarms; target level parameters; and the identity of the station (HMI/HOS) currently controlling the system. The fill pump, drain valve and mode of operation can be controlled using the HMI (see Figure 6).

2. PLC Rack

The PLC, as part of the AFTL, is the data engine that continuously serves the HMI by updating the status of all sensors and executing commands as they are received. The PLC CPU constantly cycles through all the status bits stored locally in memory including the FLSD input status and HMI pump and valve commands. The PLC will respond to changes in the state of the active memory bits in accordance with the Ladder Logic program that is stored onboard the PLC. When logic conditions are met to cause an output the PLC will execute those commands by signaling the appropriate I/O module to change the bit at an address controlled by the I/O module. The change in status by that bit will cause all monitoring programs to update. In the case of the AFTL when a command from the HMI to start the pump is received the PLC will refer to the Ladder Logic and change the status of the pump to running. All indicators that are monitoring the pump status will be updated as to its new state. For details on the PLC configuration, see Chapter III.

3. HMI Override Station

The HMI override station (HOS) is a station directly connected to the PLC that provides an alternate point of control, without intervention of the HMI. The HOS provides a set of sensors and actuators to monitor and control the system, manually. In the event the control segment were to malfunction or when local control is desired, the HOS allows an operator to take control from the HMI and force the system into a manual control mode. While in this mode, the local operator controls the system from the HOS. In MCS, a local control station like the HOS is the preferred mechanism for controlling machinery when in emergency scenarios, due to its adjacency to equipment and direct connection to the PLC (see Figure 5).

4. Automatic Mode of Operation

While in the automatic mode of operation, the AFTL operates without operator intervention. In this mode, the PLC monitors sensors and initiates commands to correct the system status when observed to be outside a target state (i.e., by controlling the fill pump and drain valve). The operator at the HMI cannot control the pump or valve, but can adjust the alarm set points except during an alarm condition.

5. Manual Mode of Operation

The HOS and HMI can each operate the system manually. While in the manual mode of operation, an operator has full control of the fill pump and drain valve from either the HMI or HOS station. When the system has completed the transition from one mode to another mode of operation, the pump and valve commands issued in the exited mode are ignored. While the system is in manual mode, the only commands accepted are those at the station in manual control (HOS or HMI).

B. DESIGN

In this section, we describe each component reflected in the CONOPS and the functionality needed by those components to support the AFTL requirements. The functionality of each component is outlined below, but implementation details are deferred to later in this chapter. The allocation of functional requirements (from Table 1) to each AFTL component is provided below, for traceability.

1. Controlling and Monitoring the Fill Pump

At the HMI, a representation of the pump control and status is presented to the operator via a graphical user interface (GUI). The operator can click to start and stop the pump. The status of the pump is displayed to the GUI, providing feedback that the command was successful.

At the HOS, the pump control is an electrical switch that signals the PLC to change the status of the pump. The pump status is displayed as a light indicator, providing feedback that the command was successful.

Requirement traceability: R2, R3

2. Controlling and Monitoring the Drain Valve

At the HMI, a graphical representation of the drain valve status and control is displayed to the operator. The operator can place a drain valve in the OPEN or CLOSED position. The status of the valve is displayed to the GUI, providing feedback that the command was successful.

At the HOS, the drain valve control is an electrical switch that signals the PLC to change the status of the drain valve to the OPEN or CLOSED position. The pump status is displayed as a light, providing feedback that the command was successful (see Figure 2).

Requirement traceability: R4, R5

3. Reporting the Status of the Tank Fluid Level

The AFTL requirements do not require devices that hold or manipulate actual fluid resources. Instead, electrical components providing continuous signals are used to implement a notional tank. The fluid level simulator dial (FLSD) is an electrical rheostat located at the HOS that can be manually adjusted to any level between 0 and 100 percent as indicated on the FLSD or 0 and 32700 gallons as displayed by the HMI (see Figure 5).

The FLSD induces electrical signals, acting in the role of a tank equipped with a level sensor. The FLSD state (referred to as "fluid level" in the remainder of this work) is displayed at the HMI graphically, and at the HOS by reading the state of the dial pointer. The FLSD position is an analog input affecting the system state (i.e., level alarms and automatic mode logic (see Figure 5).

Requirement traceability: R10 number headings

4. Dynamically Monitor the Fluid Level of the Tank

At the HOS, the operator can observe the fluid level via a physical needle pointer located on the dial using a 0-100 scale at the FSLD. At the HMI, the fluid level is displayed through an intuitive graphical interface consisting of a representation of a tank along with a numerical display of the gallons of fluid currently in the tank. In normal operation, when the fluid level is between the low and high alarm target limits, the fluid level is shown in blue. When the fluid level is below the low alarm target level, the fluid will be indicated in red. When the fluid level is above the high alarm target set point, the remainder of the tank graphic will fill with red. During normal operation the actual fluid level in gallons will be displayed next to the tank. The numerical level in gallons will be displayed next to the tank and constantly updated regardless of any alarm condition.

Requirement traceability: R1

5. Controlling the Fill Pump and Drain Valve in Automatic Mode

At the HMI, the operator uses a graphical representation of a switch to place the system in automatic mode. While in this mode, the PLC responds to the low and high level alarm conditions to control the fill pump and drain valve operations and ultimately maintaining the desired fluid level. The FLSD can be adjusted manually to simulate the effect of the tank being filled or drained.

At the HOS, a control for selecting automatic mode does not exist. When the HOS override switch is in the ON position, the HOS has control and the AFTL only operates in manual control mode.

Requirement traceability: R8

6. System Control and Indications

At the HMI, the station in control of the AFTL is displayed via a graphical indicator that displays the status of the system control. By default, control is assigned to the HMI. In the event the HOS has taken control, all HMI operator commands are ignored and the HMI operator will be unable to control the system. A graphical indicator displayed at the HMI will indicate that HMI controls have been overridden. The fill pump, drain valve, alarms, and tank fluid level status will continue to display the state of the system, accurately and continuously.

At the HOS, if the HMI Override switch is placed in the ON position, a signal is sent to the PLC to disable all controls from the HMI. The fill pump motor and drain valve can be adjusted accordingly. All changes to the system state made at the HOS will continue to be displayed at the HMI accurately and continuously regardless of the station in control.

Requirement traceability: R9

7. Targeted Fluid Level Adjustment and Alarm Indications

At the HMI, the operator can adjust the targeted alarm set points by manually setting the value for the high or low alarm condition in gallons. When the targeted level is reached, a visual alarm will blink and the fluid of the tank will change color indicating that the level of the tank has reached a desired level. The alarm levels can also be used to trigger the system to change states automatically. During automatic operation the target level will be used to signal the starting or stopping of the fill pump or the opening or closing of the drain valve.

The HOS has neither the ability to adjust the targeted fluid level nor the ability to operate in automatic mode.

Requirement traceability: R6

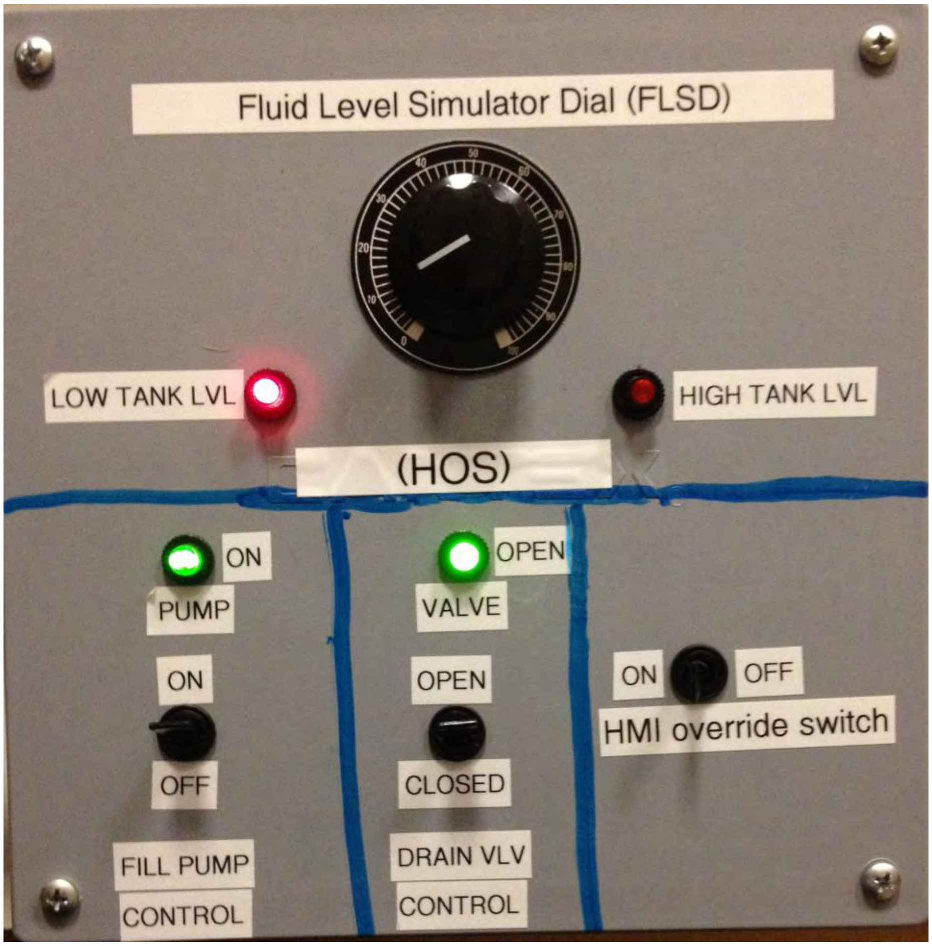
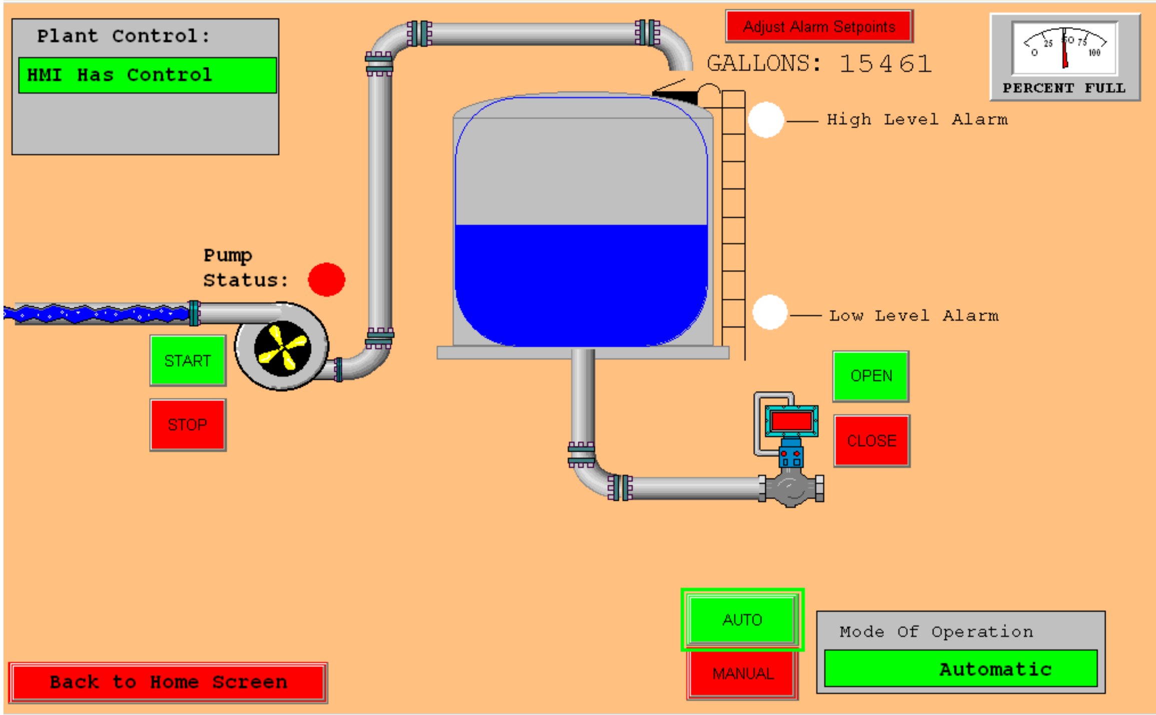


Figure 5. HMI Override Station front panel including FLSD

8. Monitoring Ethernet Traffic Within the Supervisory Segment

Packet analysis software implemented within the HMI virtual machine will allow a researcher to monitor or sample traffic sent to and from the PLC. Additionally, an Ethernet hub will be used to allow monitoring or sampling of the traffic flow between the HMI and the PLC by another machine for possible injection or exploitation purposes.

Requirement traceability: R11



Plant Control:
HMI Has Control

GALLONS: 15461

Mode Of Operation
Automatic

Figure 6. AFTL HMI screenshot (automatic mode, normal fluid level)

C. IMPLEMENTATION

Next, we describe the implementation of the AFTL, based on the various system states that the AFTL system can enter and the transitions necessary to arrive at each state. A complete state diagram is provided in Appendix D.

1. Engineering Workstation and HMI

The engineering workstation is a computer running Windows 7 as a host to a VMware virtual machine operating Windows XP SP2 as a guest OS. The host machine has two network interface cards (NIC) that allow the guest OS virtual machine (HMI) to have a direct connection to the PLC through an Ethernet connection and also connect to the campus LAN without affecting the network addressing of the NPS MCS testbed. The HMI is connected exclusively to the PLC through Ethernet on the SLC-5/05 CPU (slot 0).

At the engineering workstation, the operator uses software running on the Windows XP guest VM to implement ladder logic software and to design the HMI program. The ladder logic is uploaded to the PLC using Rockwell Automation's *RSlogix 500* programming IDE software. The HMI program is written using Rockwell Automation's *RSView32 Works* development environment and Microsoft Visual Basic [19]. During run-time, the HMI program runs under the *RSView32 Runtime* environment, which restricts changes to the layout and functionality of the HMI program. The program uses tags to store and display the sensor data and equipment status of the AFTL.

The PLC, the engineering workstation and the HMI are configured with IP static addresses but only the HMI and PLC share a subnet (see Figure 7).

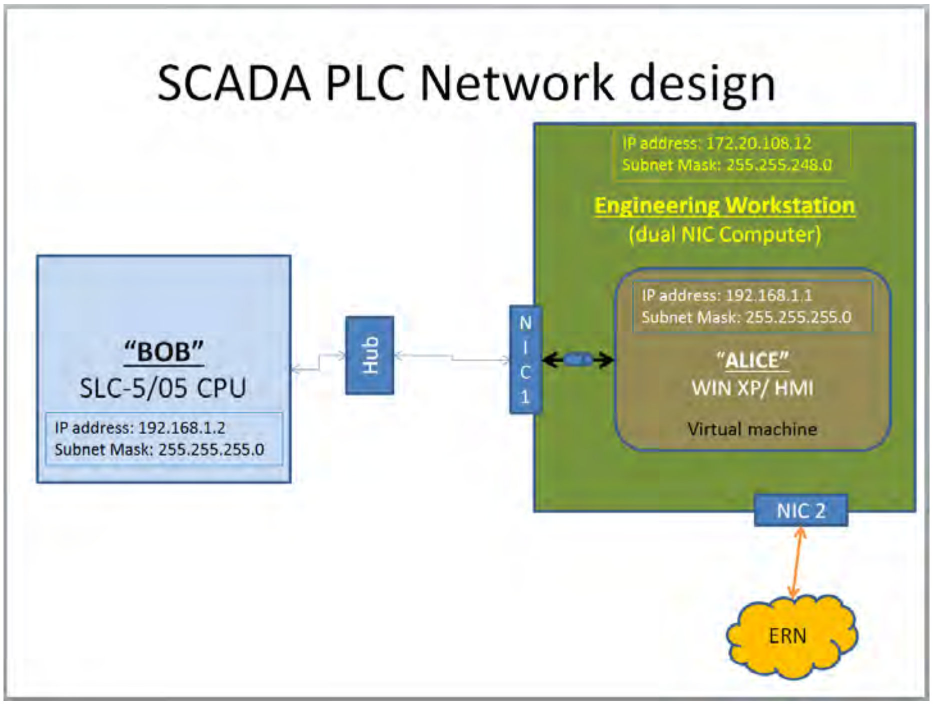


Figure 7. Testbed network design

2. AFTL PLC

The PLC program (written in ladder logic) receives HMI and HOS commands, monitors inputs for changes and outputs signals when conditions are met internally. The status of all sensors are continuously updated by the PLC CPU cycle and displayed accordingly.

The PLC must be powered up prior to HMI operation. The PLC is working properly when the following conditions are met: 1) the PLC has 120VAC power, 2) the red power LED located on the power supply is lit, and 3) the “FLT” flashes on briefly followed by all the LEDs and then they are all quickly extinguished (see Figure 8). After a successful power-up sequence, the “RUN” and “ENET” lights should be lit and the PLC programming key must be in the “REM” position (see Figure 10). The power-up sequence fails when the indications described are not met or if the “FLT” light is lit constantly [20].

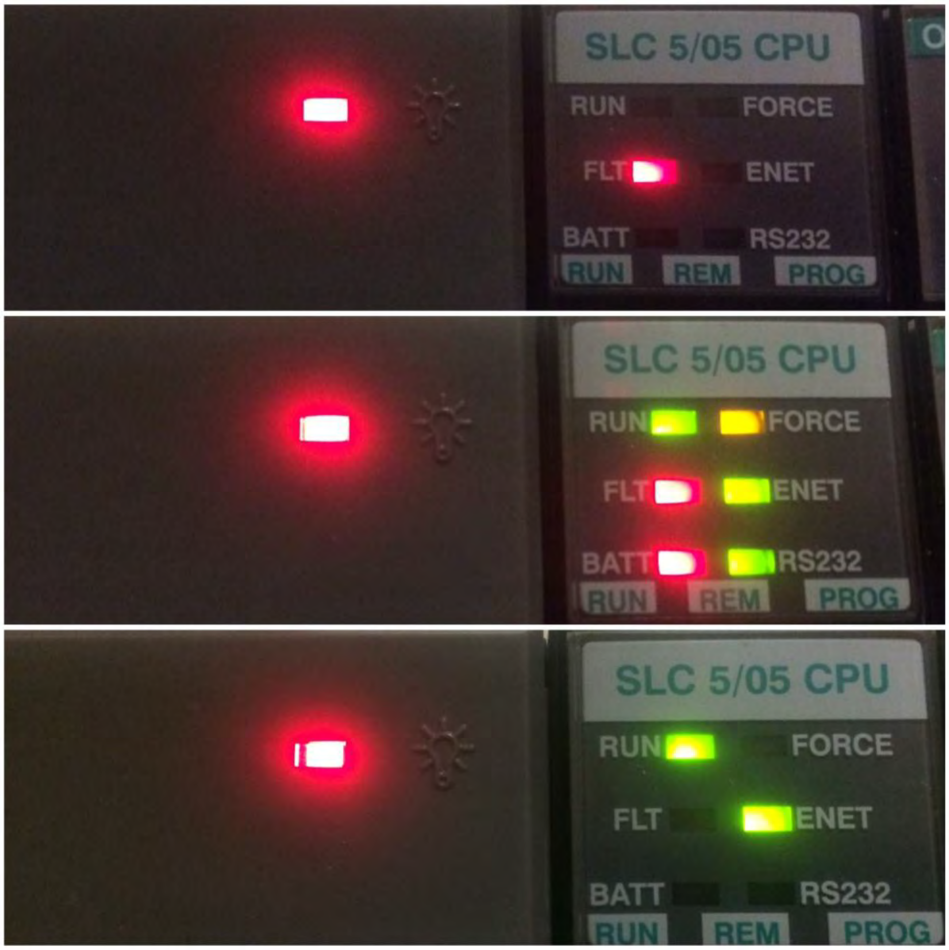


Figure 8. Sequence of indications (1–3) during power-up of the PLC

Once fully powered up, the PLC receives commands from the HMI and HOS as specified by the operator, and retrieves all data from each I/O module via the common data bus between modules. The commands from the HMI are sent via the Ethernet connection to the CPU module. Commands are issued by changing a predetermined status bit inside the PLC memory. The status bits are monitored and updated each cycle of the PLC CPU. The CPU will update the status of sensors and act on commands from the HMI and outputs at the HOS based upon the ladder logic program stored onboard the PLC.

The ladder logic program is a graphical programing language that acts as a circuit where the input to the ladder rung affects some output. Ladder logic is commonly used in PLC's in accordance with the IEC 61131-3 [21]. The ladder logic program used in this testbed is loaded onto the internal memory of the CPU module via the Ethernet connection between the PLC CPU module and the Engineering Workstation using the RSLogix 500 IDE software [20], [22]. During the upload process the PLC will no longer monitor or update the status of any sensors. Once the PLC is placed in RUN mode, the uploaded ladder logic program will start executing.

The ladder logic program code that controls the AFTL consists of a main program and four other sub-programs. Each sub-program is enabled or called by the main program and act as methods supporting the various states of the AFTL. The main program is used to enable the four sub-programs: automatic mode, manual mode, HMI control, or HOS control. The compartmentalization into isolated sub programs provides exclusive control of the AFTL in each state and prevents multiple station commands from being executed by the PLC without having control of the AFTL. By design, only the station in control can change the status of control bits inside the PLC and affect ladder logic stored there. In manual mode, at the HMI or at the HOS, the controlling station will issue a command that causes logic stored in the PLC to affect an output or change the status of an indication. In the event the AFTL is placed in automatic mode, the PLC ignores the controlling stations fill pump and drain valve commands and operates independently from the HMI and HOS. The PLC internally will invoke the sub-program that issues commands to the pump or drain valve in reaction to targeted fluid levels (as input by the FLSD) in the tank previously set by the HMI operator.

At the HMI, the HMI programs are written using "tags" as variables that allow access to data stored on the PLC. Tags are used to update the HMI graphical display data with current sensor data stored on the PLC at each cycle of the PLC CPU. Since the sensor data stored on the PLC memory is accessible regardless of which station is in control, sensor updates or tag updates are always available to the HMI. The HMI software uses tags much in the same way pointers are used in other programming languages. In the RSView32 Works IDE the tags "point" to a memory location on the PLC by addressing it directly and interpreting the raw data based on the defined data type. In this case, the data type is either "digital" or "analog" and is given a unique name or handle that can cause an animated object to change color, shape or appearance depending on the desired effect. When the HMI is in full operation all tag data are duplicated in a database stored on the local machine running the HMI software. The PLC CPU updates the data during multiple cyclic data updates per second [19].

Sensors are directly connected to the PLC using I/O modules, which perform updates to all stations in the event of a change in status of a sensor. As an example, the pump switch on the HOS is wired to provide a digital (on or off) input to the digital I/O module 1746-IO12DC at slot 3 in the PLC rack (see Figure 10). The change in status sensed at the I/O module initiates a ladder rung on the PLC, to change the voltage value in a memory location that signals the 1746-NO4V output module at slot 2 to send power to an indicator light. The same voltage value that controls the indicator light is also monitored by the HMI using a "tag" that monitors a specific parameter and in this case senses the change in voltage. The change in voltage from 0V to 24V initiates a change in the pump status with a blinking green icon at the HMI indicating that the pump is running (see Figure 9).

3. HOS

The HMI override station is a connection box with a direct electrical connection to the PLC (see Appendix B for diagram). At the HOS, a command is sent as an electrical signal in the form of a change in voltage or current to the PLC and is received by an I/O module. The I/O module will update the CPU module with the new sensor status using a shared bus located on the PLC rack and ensure all stations are updated with the status of the system control, fill pump or drain valve as changes are made at the HOS.

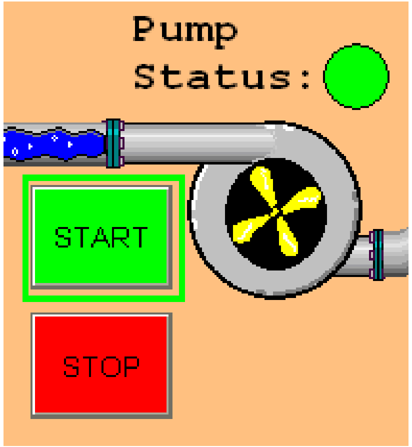


Figure 9. HMI pump status indicator and controls

The HOS has four LED lights that operate on 10VDC and three single-pole single-throw (SPST) switches that control the flow of 24VDC to the PLC. The LED lights are comprised of two red and two green lights.

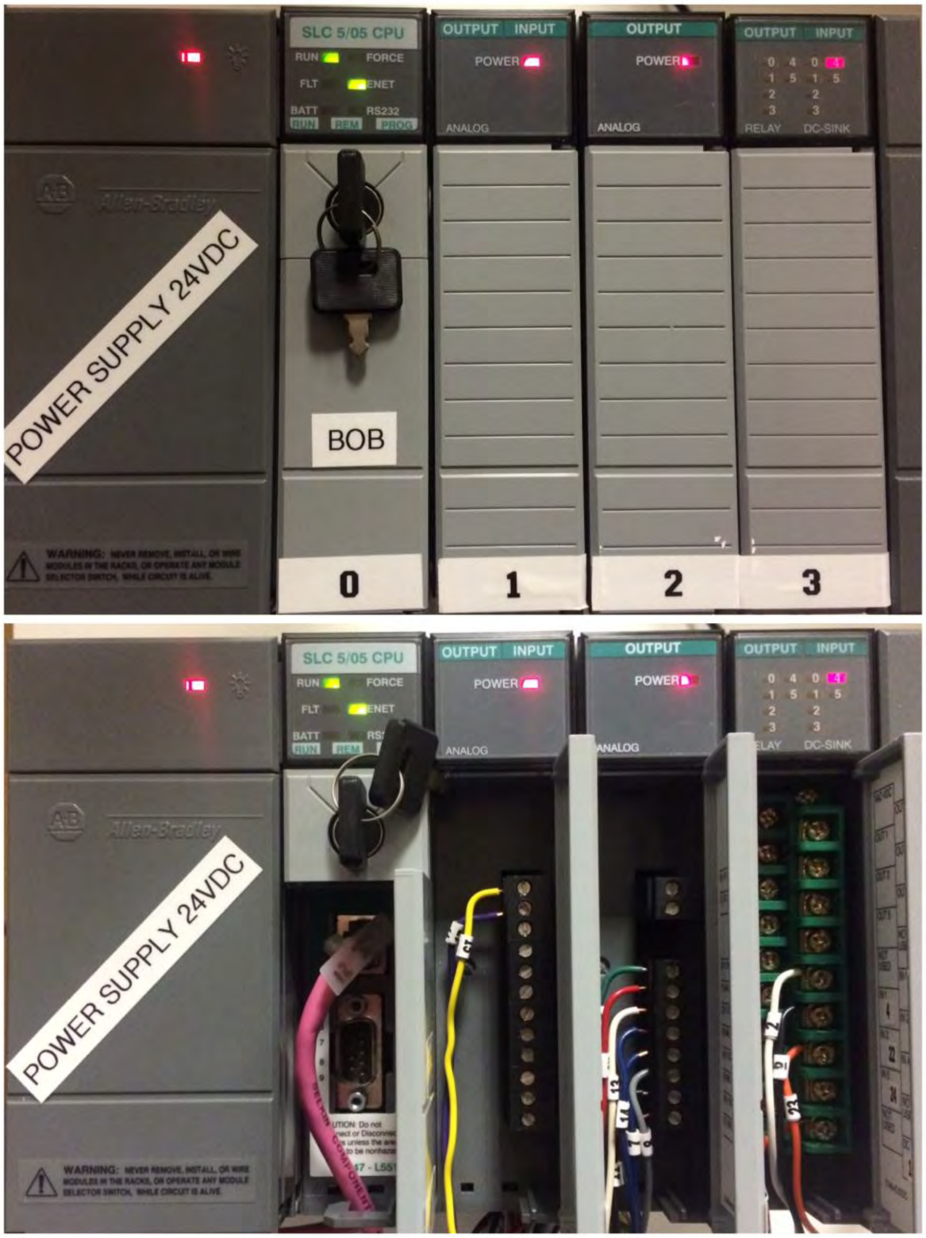


Figure 10. PLC I/O modules their slot locations 0–3 (left to right)

The two red LED lights are illuminated by a 10VDC signal from the AB 1746-NO4V analog I/O module (PLC slot 2) when a low or high level is indicated at the FLSD. The two green LED lights are indicators for the fill pump operation and drain valve operation. The green LED lights are illuminated by the AB 1746-NO4V analog I/O module (PLC slot 2) with 10VDC in the event that the ladder logic conditions have been met by the HMI or the HOS to order the pump or valve to turn on. The indication for the pump or valve is strictly controlled by the PLC and will indicate at the HOS regardless of the station that is in control.

The three electrical switches on the HOS are used for taking control of the AFTL at the HOS, commanding the fill pump on or off and commanding the drain valve open or closed as desired (see Figure 11).

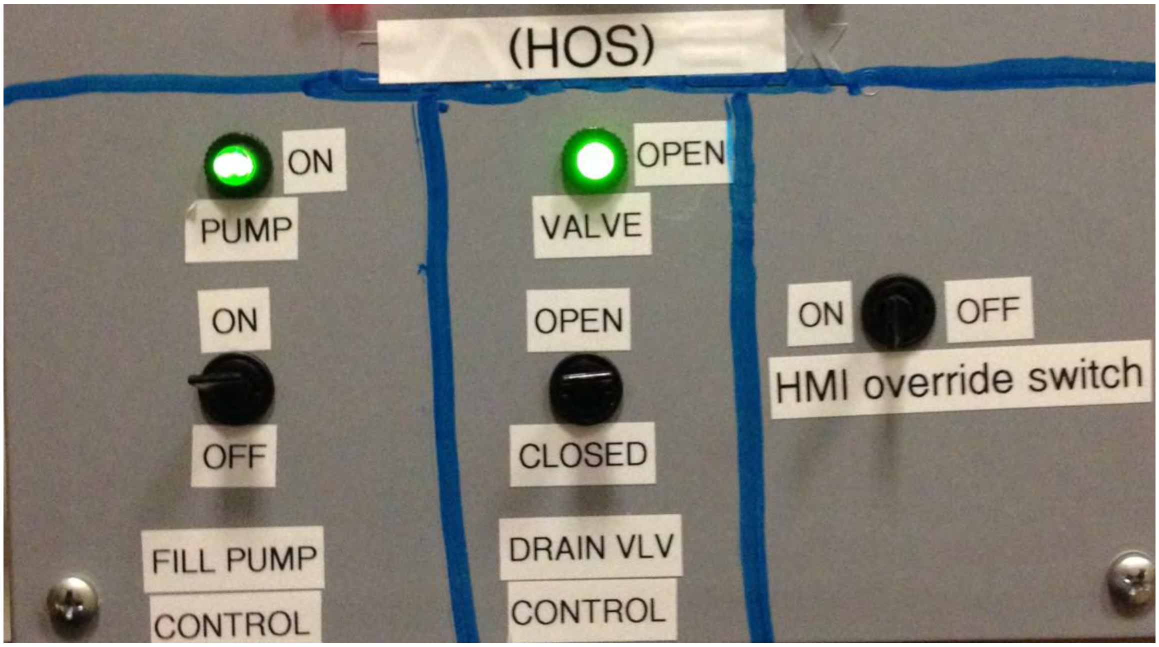


Figure 11. HMI override station controls and indications

The HMI override switch controls a 24VDC electrical signal that is sent to the PLC signaling control override of the system in the event of an emergency or when only manual local control is desired. The remaining two switches operate the same way but the PLC interprets the change in electrical voltage as pump and valve controls where 0 VDC is pump off or valve closed and 24VDC is pump on or valve open.

THIS PAGE INTENTIONALLY LEFT BLANK

V. DIGITAL I/O LAB

The digital I/O lab is the sub-component of the MCS testbed that provides a platform allowing others to experiment with MCS systems using Digital I/O. The ability to analyze the data packets and compare between input command and output response is a critical aspect of the digital I/O lab.

A. CONCEPT OF OPERATIONS

The digital I/O lab resembles an authentic control system that interacts with a third-party actuator through a specialty digital protocol. Functional requirements of the digital I/O lab are listed in Table 2. The digital I/O lab consists of four components: the HMI (supervisory segment), the PLC (control segment), mechanical (robot) arm and proximity sensors. The robot arm moves an object between sensors, as commanded. The proximity sensors detect the location of the object and update their status to the PLC digitally. The HMI will interface with the PLC and display the status of the sensors and the robot arm to the HMI operator. The HMI operator can command the arm to move an object between sensors and be informed of the condition of that operation.

1. Digital I/O Human Machine Interface

The digital I/O HMI is a graphical display of the sensors and status available in the lab. The digital I/O HMI allows the operator to command the arm to move an object between proximity sensors, to view the current command being sent to the arm and to view alerts that may need the operator’s attention (see Figure 12).

2. Programmable Logic Controller

The PLC in the digital I/O lab is the same described earlier (see Chapter III). The digital I/O lab will utilize a separate I/O module than that of the AFTL, but the concept of operation is the same.

3. Robot Arm

The robot arm resembles devices used in a manufacturing setting, and fills the role of a third-part actuator requiring communication of specific commands sent from the controlling station (HMI) and the output device (robot arm). The robot arm is a servo-driven commercial arm that is directly connected to the PLC CPU module using a RS232 serial connection. The robot arm is controlled using a pre-formatted command signal that is sent from the HMI. The robot arm can be used to move an object from one location to another. Each location has a proximity sensor that will provide feedback to the PLC to determine if the command to the robot arm is successful.

4. Proximity Sensors

Proximity sensors are used to detect the presence of the object moved by the robot arm. The proximity sensors have a digital output that signals the PLC when the object is placed on the sensor or removed from the sensor. The sensors state is used to determine the current status of the digital I/O lab and the completion status of the command issued to the robot arm. Proximity sensors are used quite frequently in industrial settings and assembly line systems.

B. DESIGN

The design section will describe each component of the digital I/O lab listed in the CONOPS section above and the functions needed by each component to support the functional requirements of the digital I/O lab (see Table 2). The functionality of each component is outlined below, but implementation details are deferred to later in this chapter. The digital I/O lab shares the Allen Bradley PLC used in the AFTL, but utilizes the RS232 port on the CPU card to send commands to the robot arm and requires different I/O modules to monitor the status of the proximity sensors influenced by the robot arm. The digital I/O network shares the same components and network as described in Chapter IV. See Table 2 for functional requirements associated with the components discussed below.

1. Controlling the Robot Arm

At the HMI, the robot arm is controlled using command buttons and feedback from proximity sensors. Each proximity sensor and its status are represented graphically at the HMI. There are two buttons displayed as options at each proximity sensor when it is active. These buttons represent the choice of movement that the robot arm can perform. Each button will execute the commands the robot requires to move an object from one proximity sensor to another. As the robot is in motion, each command that is transmitted to the robot arm is displayed on the HMI. The HMI operator cannot send new commands to the robot arm until the robot arm has completed the task of moving the object to the new proximity sensor.

The robot arm can be commanded to a starting and stowing position using the start and stop buttons located on the HMI. The start button will place the robot arm in a position ready to retrieve and move an object. The stop position will order the robot arm to shut down completely and wait for a start command. The stop position is the offline position for the robot arm when the digital I/O lab is not in use.

Requirement traceability: R12

2. Moving Objects Using the Robot Arm

At the HMI, each proximity sensor status is displayed and if an object is present on the proximity sensor the HMI will represent that status. When an object is present on the proximity sensor the HMI will provide choices to the user in the form of buttons. Each button will allow the operator to move the object between proximity sensors as desired. Once the command is given to the robot arm to move the object between sensors then the robot arm will perform the movements necessary to pickup and deliver the object onto the desired sensor.

Requirement traceability: R12

3. Dynamically Displaying the Status of the Proximity Sensors

At the HMI, A graphical representation and a text status will be displayed for each of the three proximity sensors. When a sensor detects an object, the status of the sensor will change color indicating that the sensor detects an object. Likewise if the sensor no longer senses an object it will indicate as such.

Requirement traceability: R13

4. Displaying the State of the Robot Arm and Commands Issued

At the HMI, The robot arm has three main positions that are tracked: start, stop and in motion. The exact status of the command issued to the arm is displayed as an ASCII representation of the bits used to issue the command to the robot arm. Once the arm receives each command, the next command will be displayed until the robot arm has completed the movement. The robot will return to the start position upon completion of the movement.

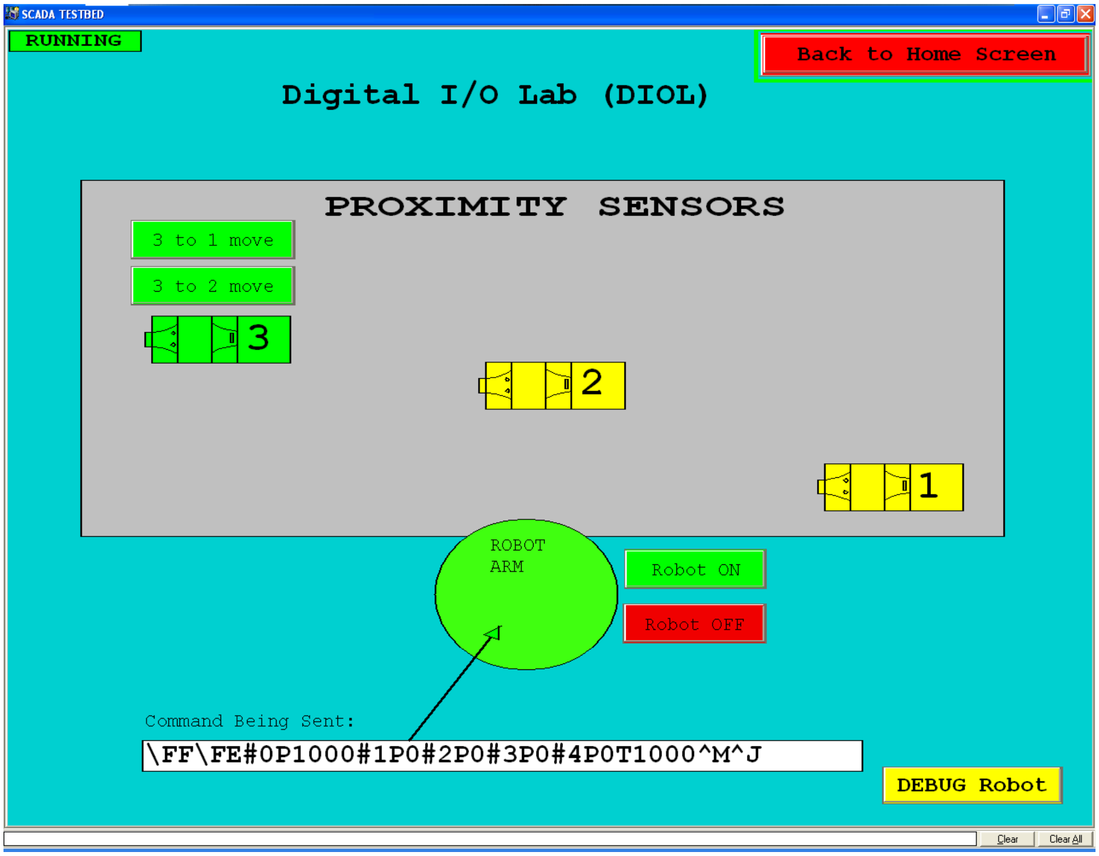
At the robot arm, the movement of the arm can be observed and verified visually by the HMI operator to ensure its proper execution.

Requirement traceability: R14, R15

5. Displaying Indications When an Error has Occurred

At the HMI, when all three proximity sensors indicate no object the HMI will indicate there is no object present and prompt the operator to replace an object on a sensor to continue. Additionally when all three sensors have objects on them the robot arm cannot move to an empty sensor and therefore an error will be displayed by the HMI indicating that all the sensors are full.

Requirement traceability: R16



Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Figure 12. DIOL HMI screenshot

C. IMPLEMENTATION

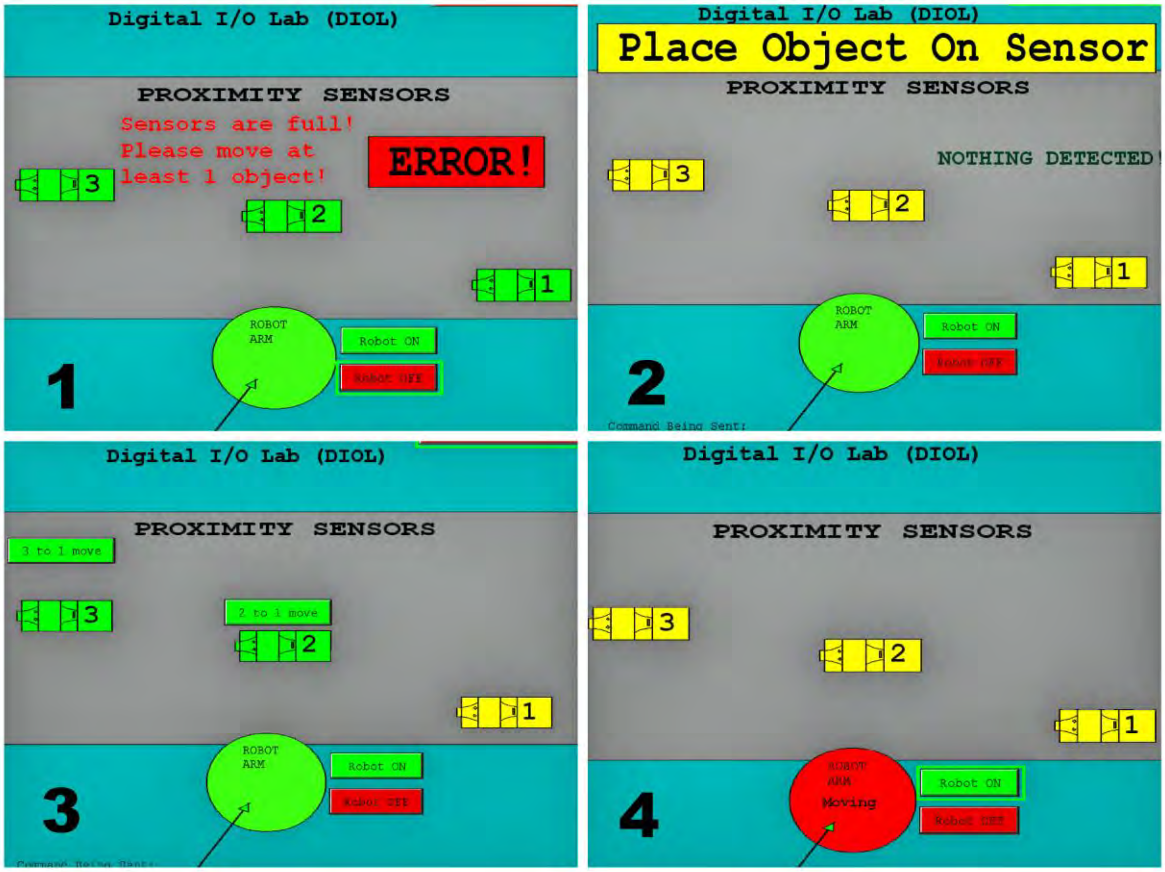
Implementation of the Digital I/O lab design described earlier in the chapter is described in this section in detail. A state diagram for the DIOL can be found in Appendix E.

1. Engineering Workstation and HMI

The engineering workstation used in the Digital I/O is the same as that described in the AFTL. The HMI is implemented using graphical buttons and colors to display the status of the robot arm and each proximity sensor. The DIOL HMI uses Visual Basic programs that read a text file of pre-formed commands to execute movements between sensors. Logic is used at the HMI to decide if a button is available to be displayed or changes color based on the status of each proximity sensor and the robot arm. The DIOL

HMI display includes a "DEBUG" section from which the operator can send commands directly to the robot arm, bypassing the Visual Basic program. The debugging capability allows the operator to perform robot movements one at a time to ensure the robot execute each command properly.

In the event of an error involving the proximity sensors or the robot the HMI has been programmed to identify the errors to the operator. An example of the errors can be found in Figure 13, which shows the various proximity sensor states. Figure 13 is labeled as follows: (1) Error: all three sensors active robot arm cannot move any of the three objects, (2) Error: No object for the robot arm to move because none of the sensors are active and robot arm is not in motion, (3) Normal: 2 sensors active, only choice for the robot arm to move is to sensor 1, (4) Normal: robot arm in motion with object. See Figure 12 for a screenshot of the DIOL HMI and Chapter IV for details on the HMI and engineering workstation.



Sensors are full!
Please move at
least 1 object!

Figure 13. DIOL HMI, various proximity sensor display states

2. Digital I/O PLC

The PLC in the digital I/O Lab is the same used in the AFTL lab (see Chapter III). The digital I/O lab requires only one I/O module and channel 0 output on the PLC CPU module to retrieve and update information about the proximity sensors and send commands to the robot arm. The CPU module output channel 0 interfaces with the robot arm using the built in RS232 port and cable connection located on PLC CPU module. The proximity sensors are wired directly to the digital I/O module located in slot 3 of the PLC. The PLC receives HMI commands and can issue commands to the robot arm based on the internal ladder logic program. The operation of the ladder logic program, interface between the HMI and PLC and HMI operation using “tags” are identical to those described previously (see Chapter IV).

While the interface and basic operation of the HMI, PLC and sensors are the same in the digital I/O lab; the ladder logic program used to operate the robot arm and update the status of the three proximity sensors is different. The ladder logic is composed of a main program and two subroutines. The main program enables the two subroutines when the digital I/O lab HMI is enabled. One subroutine monitors the proximity sensor for change in status and the other interfaces with the robot arm to transmit commands received from the HMI. When the Digital I/O Lab HMI is open, the PLC will constantly monitor the proximity sensors for status changes and allow the HMI operator to transmit commands to the robot arm via channel 0 on the PLC CPU using the RS232 port.

3. Robot Arm and Proximity Sensors

The robot arm is a Lynxmotion AL5D mechanical arm with four servos, connected to a SSC-32 servo controller and interface card. The SSC-32 servo controller interface receives commands from the PLC on the RS232 connection and executes them. The four servos provide range of motion in three dimensions, and a clamping action on the hand of the robot arm. The servos have rotors that will rotate to a position as commanded. Each servo is numbered and can be addressed in order to command it to move. The servos are commanded to a position by an integer that represents where on the scale of motion the servo should place its rotor. Once the command is received, the SSC-32 servo controller will compare the current rotor location to the commanded rotor location and move the rotor until the difference between the ordered and actual position is zero. The difference between the ordered and actual position of the rotor is called the error [23].

The servo controller commands are formatted in the following pattern: “#0P1000#1P1500#3P2000T1000/n”. Where the integer following every “#” represents the servo to be addressed. The servos are numbered from the base to the hand clamp as 0–4 respectively. For example, “#1” addresses the number 1 servo. Immediately following the servo address is the letter “P,” which represents a command for the ordered position to place the rotor. The integer following “P” is the numerical representation of the rotor location. For example, the command “#1P1000” is interpreted as a request to place servo rotor “1” at position “1000”. The last portion of the command used to move the robot arm is the “T” command. The “T” command indicates the time in milliseconds that it should take to move each servo rotor to the ordered position such that the error is zero. Additionally, each servo can be addressed and given a position command in a single command. For example, the previous command “#0P1000#1P1500#3P2000T1000/n” requests to place the rotors of servos 0, 1 and 3 to the numerical positions indicated in 1000 milliseconds (1 second). The carriage return “/n” communicates the end of the command and orders the servo controller to execute the request. After this example command is executed, the “0” servo will be at position 1000, the “1” servo will be at position 1500, the “3” servo will be at position 2000, and the command should take no longer than 1 second to execute. At the HMI, the commands sent to the PLC are displayed in the ASCII representation, as in the examples above, to allow the HMI operator to easily verify the command being executed by the robot arm. The PLC sends each command in binary format to the Lynxmotion SSC-32 servo controller which in turn orders the movements of the servos on the robot arm as required by the command [24].

The proximity sensors are capacitance-sensitive relay sensors manufactured by the TURCK Company. The proximity sensor internal relay will change state from normally open to closed position when they sense a large change in density within 2mm of the sensor [25]. The change in density triggers the sensor relay to close and indicate to the PLC the presence of an object. The HMI will indicate this change as updated by the PLC.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. TESTING

The test plan ensures that each state in the AFTL and DIOL state diagrams (described in Appendices D and E) is achievable under normal conditions and verify that each design requirement is met. It includes both functional tests and exception tests, described below.

A. FUNCTIONAL TESTING PLAN

The purpose of functional testing is to verify the functionality of each AFTL and DIOL component under normal conditions. Table 3 summarizes the AFTL and DIOL functional tests. A designator of the form TF*n* identifies each test, where *n* is a unique number.

Table 3. Functional requirements

Test ID	**Functional Test Description**	**Requirement Tested**
TF1	Test the display of the fluid level using the FLSD	R1, R10
TF2	Observe, start and stop the fill pump	R2, R3
TF3	Observe, open and close the drain valve	R4, R5
TF4	Tank level alarm operation	R6
TF5	Tank level alarm set point adjustment	R7
TF6	Automatic mode operation	R8
TF7	Override control functionality test	R9

Test ID	Functional Test Description	Requirement Tested
TF8	Ethernet traffic monitoring	R11
TF9	Move objects between proximity sensors using the robot arm	R12, R13, R17
TF10	Properly display proximity sensor status and logic	R13, R17
TF11	Properly display robot arm status	R14
TF12	Properly display robot arm commands	R15

The objective and expected result of each functional test are described below.

TF1

- Description: Verify the proper display of the fluid level using the FLSD.
- Requirement Tested: R1, R10
- Expected Result: At the HMI, the fluid level and numerical displayed should both move as the level is adjusted.

TF2

- Description: Verify the ability to observe, start and stop the fill pump in both normal and HMI-override modes.
- Requirement Tested: R2, R3
- Expected Result: The HMI and HOS displays should indicate the status of the fill pump as each command is executed

TF3

- Description: Verify the ability to observe, open and close the drain valve in both normal and HMI-override modes.
- Requirement Tested: R4, R5

- Expected Result: The HMI and HOS displays should indicate whether the valve is open or closed after each command execution.

TF4

- Description: Verify the detection and handling of tank level alarms using the FLSD.
- Requirement Tested: R6
- Expected Result: The HMI and HOS should display an indication of an alarm when the fluid level reaches either the low and high alarm set point.

TF5

- Description: Verify the ability to adjust the tank level alarm set points.
- Requirement Tested: R7
- Expected Result: The system should return to normal operation without any errors after the adjustment of the alarm set points. The HMI display should show the new alarm set points for the High and Low levels within the tank graphic.

TF6

- Description: Verify the ability to control the fill pump and drain valve while operating in the automatic mode.
- Requirement Tested: R8
- Expected Result: The HMI should indicate the mode of operation to be "Automatic". When the tank level is "Low" the fill pump should start and drain valve should indicate closed. When the tank level is "High" the drain valve should open and the fill pump should remain off.

TF7

- Description: Verify the ability to switch to the HMI-override mode and to control the fill pump and drain valve while in this mode.
- Requirement Tested: R9
- Expected Result: The AFTL HMI "Plant Control" indicator should indicate "Override Station," which confirms that the HOS has control of the AFTL. The HOS operator should have the ability to control the fill pump and drain valve as well as have status indications for both.

TF8

- Description: Verify the ability to monitor Ethernet traffic between the control (HMI) and supervisory (PLC) network segments.
- Requirement Tested: R11
- Expected Result: The engineering workstation operator will have the ability to intercept Ethernet traffic between the control (HMI) and supervisory (PLC) network segments and conduct analysis as needed.

TF9

- Description: Verify the ability to operate the robot arm and move objects between proximity sensors using the robot arm.
- Requirement Tested: R12, R13, R17
- Expected Result: The Digital I/O lab HMI should move up to two objects as commanded and display the correct status of all proximity sensors after each command.

TF10

- Description: Verify the ability to display proximity sensor status as objects are manually placed on the sensors.
- Requirement Tested: R13, R17
- Expected Result: The DIOL HMI should display all proximity sensor status changes as objects are put on and removed from the sensors.

TF11

- Description: Verify the ability to display the status of the robot arm's movements.
- Requirement Tested: R14
- Expected Result: The Digital I/O lab HMI should display whether the robot arm is moving or not moving both graphically and textually.

TF12

- Description: Verify the ability to display robot arm commands as they are issued.
- Requirement Tested: R15
- Expected Result: The Digital I/O lab HMI should display all commands sent to the robot arm for verification and troubleshooting purposes.

B. EXCEPTION TESTING PLAN

Exception testing will perform a task that is outside of normal operation or parameters and provide the expected response by the system. The testbed should be able to handle all expected exceptions and continue operating. Table 4 summarizes the exception tests. A designator of the form TE*n* identifies each test, where *n* is a unique number.

Table 4. Exception tests

Test ID	**Exception**	**Associated Requirement(s)**
TE1	Tank level alarm value not within acceptable range or format	R7
TE2	Illogical tank level alarm values	R7
TE3	Robot arm fails to move or deliver object to sensor	R12, R16
TE4	Robot arm disturbs wrong object	R12, R16
TE5	No proximity sensor active	R13
TE6	Unable to move objects with robot arm, no open sensors exist	R13

The objective and expected result of each exception test are described below.

TE1.

- Exception: Tank level alarm value is not within acceptable range or format; while performing functional test TF5, value entered by the operator is not within the 0 to 32700 gallon range or not the proper format.
- Associated requirement: R7
- Expected result: The text boxes should turn red and an error will be displayed indicating unacceptable values were entered.

TE2.

- Exception: Invalid tank level alarm values; while performing functional test TF5, the low alarm value entered by the operator is greater than the high alarm value.
- Associated requirement: R7
- Expected result: An error will be displayed indicating the values entered are not valid.

TE3.

- Exception: Robot arm fails to move or deliver object to the specified sensor; while moving an object in functional test TF9, the robot arm drops the object or the object does not arrive at the destination proximity sensor
- Associated requirement: R12, R16
- Expected result: An error will be displayed prompting the operator to replace the object on the proximity sensor. The error message will indicate on which sensor the object should be replaced.

TE4.

- Exception: Robot arm unintentionally knocks a stationary object (an object not ordered to be moved) off of a proximity sensor while moving an object.
- Associated requirement: R12, R16
- Expected result: An error will be displayed prompting the operator to replace the object on the proximity sensor. The error message will indicate on which sensor the knocked off stationary object should be replaced.

TE5.

- Exception: No proximity sensor active after the DIOL enters the Ready State while performing functional test TF10.
- Associated requirement: R13
- Expected result: An error will display at the HMI indicating no object is present.

TE6.

- Exception: Unable to move objects with robot arm, all proximity sensors are full and no open sensors exist;
- Associated requirement: R13
- Expected result: an error will display at the HMI indicating that all three proximity sensors are occupied. The HMI error will prompt the operator to remove one of the objects.

C. TESTING PROCEDURE

Full testing procedures are provided in Appendix F.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION

This thesis described the construction of the NPS MCS security lab, including its motivation, design, implementation, and testing. The testbed helps address the need for better research and testing in the cyber security domain, to help prepare the U.S. Navy for modern cyber security threats facing industrial control systems and machinery control systems. Allowing researchers to experiment with authentic physical systems provides insight into the security challenges in ICS and can help validate otherwise difficult-to-test scenarios. Additional research in testbed design as well as future experimentation on this testbed will benefit the U.S. Navy and the entire cyber security community, and may lead to strengthening critical infrastructure used within the military and civilian industry.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. TESTBED WIRING SCHEMATIC

The testbed wiring diagram schematic (see Figure A1) shows the connections between the PLC and various components AFTL and DIOL in a conceptual level in order to understand what is required to make each part operate. The exact locations of the connections are described in Appendix B and C under the AFTL and DIOL wiring diagrams.

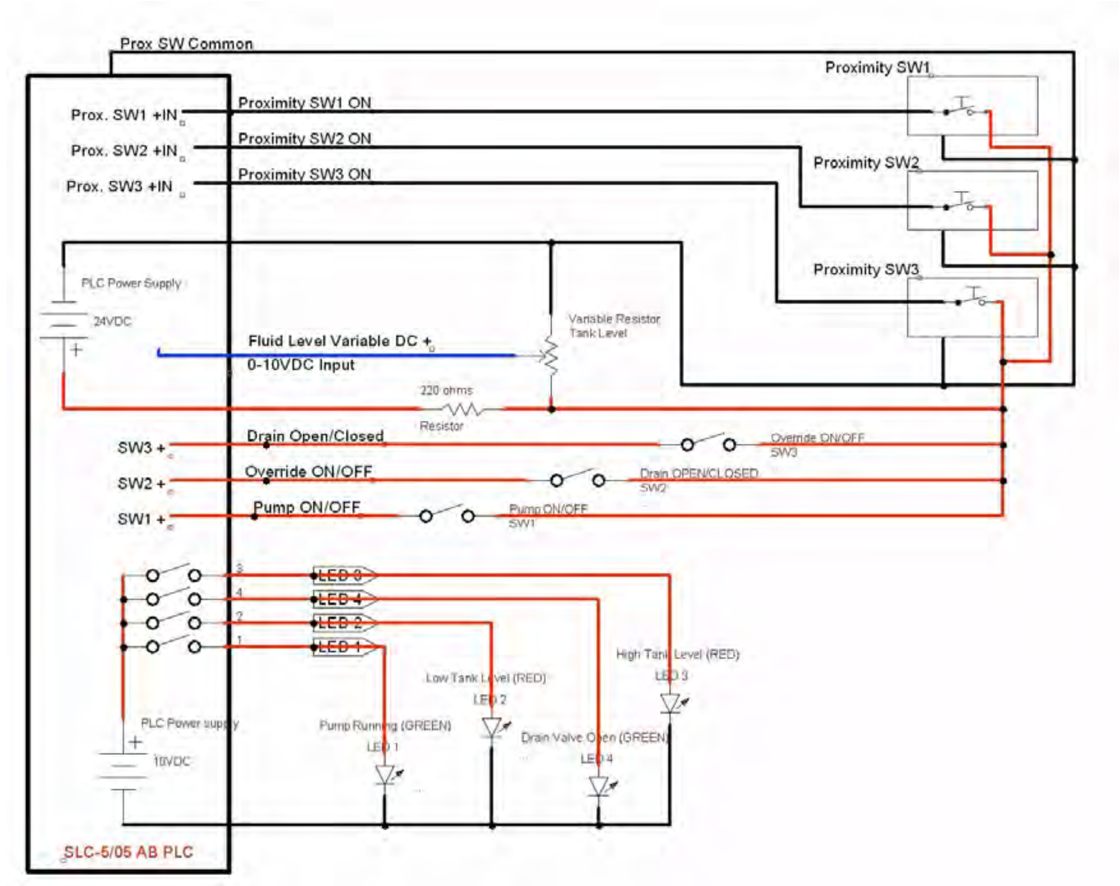
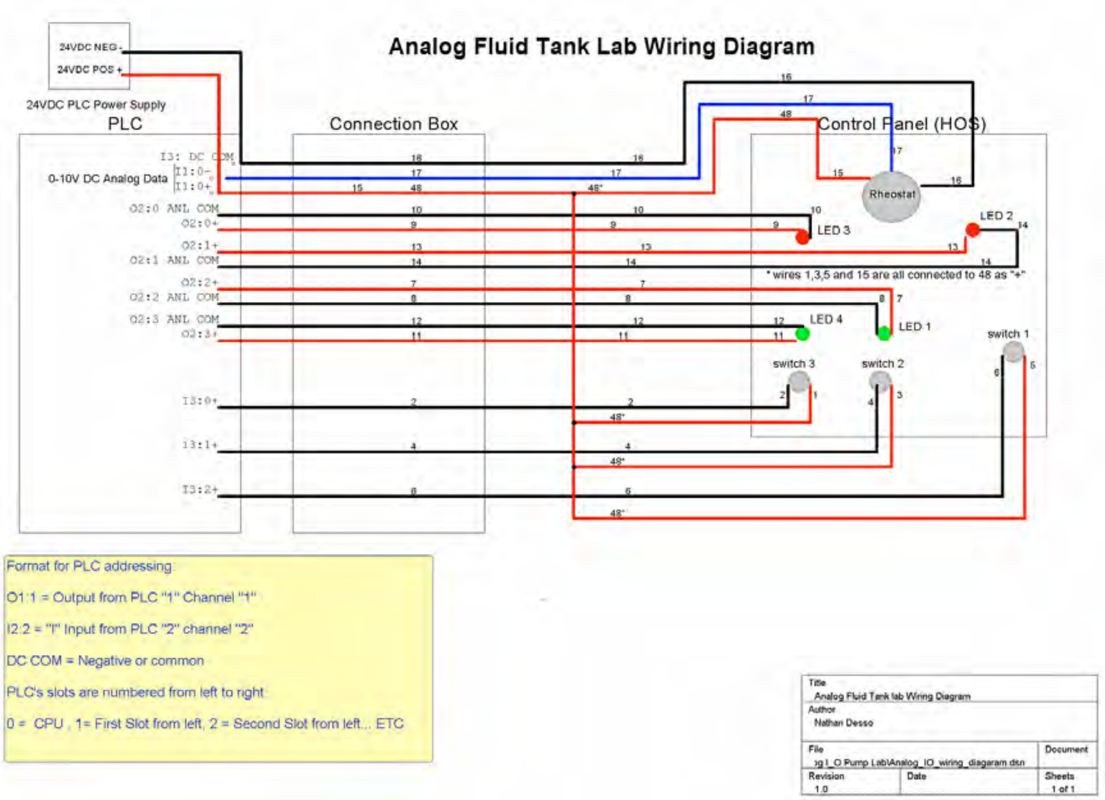


Figure A1. Electrical Schematic, Analog Fluid Tank Lab

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. AFTL WIRING CONNECTION DIAGRAM

The AFTL wiring diagram shown in Figure B1 provides the exact connection locations for each component in order to replicate assembly or to troubleshoot.



Format for PLC addressing:

O1:1 = Output from PLC "1" Channel "1"

I2:2 = "I" Input from PLC "2" channel "2"

DC COM = Negative or common

PLC's slots are numbered from left to right

0 = CPU , 1= First Slot from left, 2 = Second Slot from left... ETC

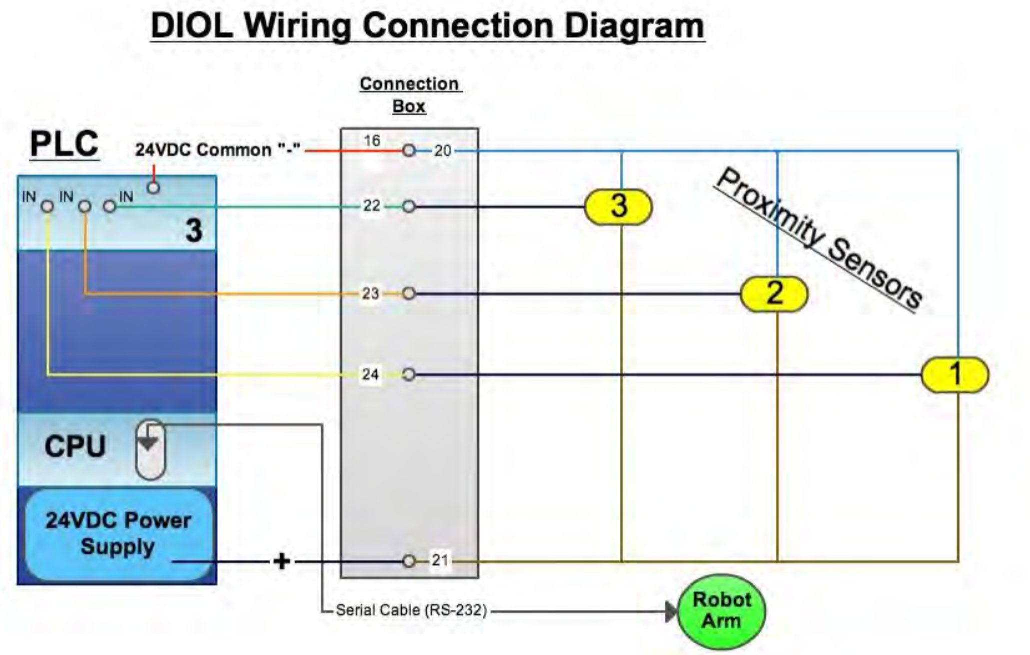
Title Analog Fluid Tank lab Wiring Diagram		
Author Nathan Desso		
File ɔg I_O Pump Lab\Analog_IO_wiring_diagaram.dsn		Document
Revision 1.0	Date	Sheets 1 of 1

Figure B1. AFTL wiring diagram

THIS PAGE INTENTIONALY LEFT BLANK

APPENDIX C. DIOL WIRING CONNECTION DIAGRAM

The DIOL wiring diagram (see Figure C1) shows the connections needed to assemble the proximity sensors and robot arm to the PLC. The DIOL wiring diagram provides the exact connection locations for each component in order to replicate assembly or to troubleshoot.

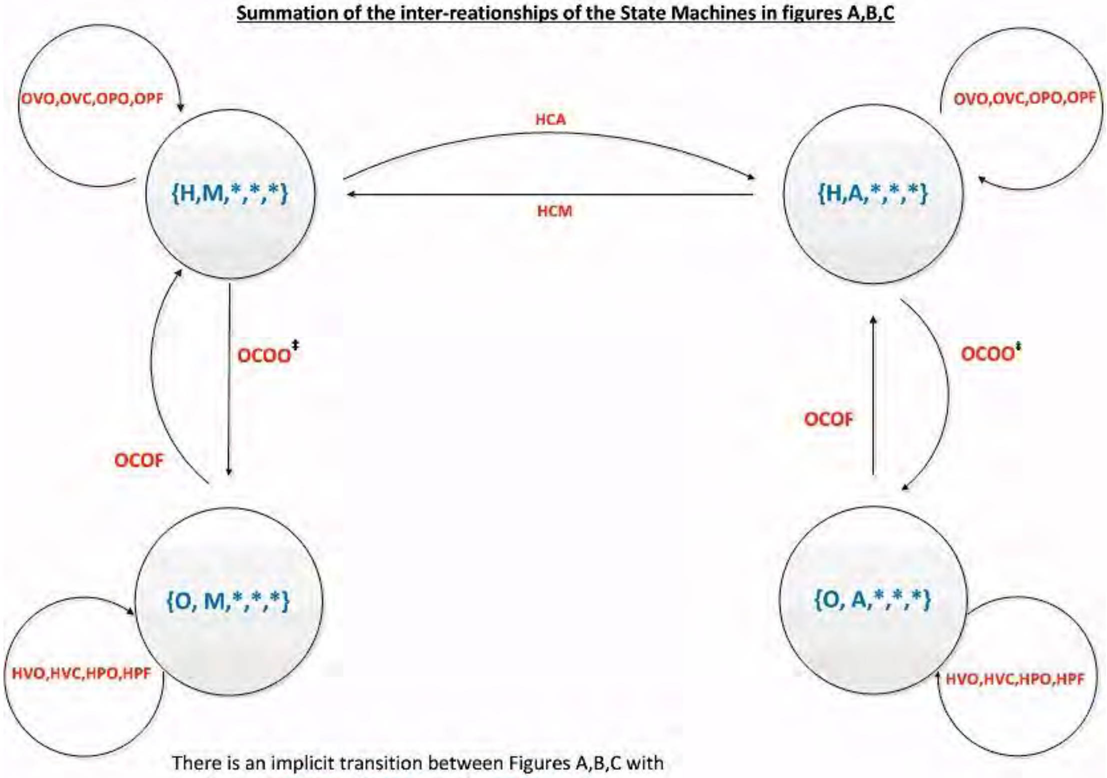


C1. DIOL wiring connection diagram

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. AFTL TESTBED STATE DIAGRAM

This appendix presents the state diagrams for the AFTL. Each state is represented as a 5-tuple format: {A,B,C,D,E}, where "A" represents the station in control, "B" is the mode of operation of the HMI (Manual or Automatic), "C" represents the status of the fill pump, "D" represents the status of the drain valve and "E" represents the status of the level alarms. For example: {H,M,P,X,(HTLN,OTLN)} represents the state when the HMI is in control, the HMI is in manual operation mode, the pump is running, the valve is closed and there are no high or low alarms (normal tank level).



Summation of the inter-reationships of the State Machines in figures A,B,C

There is an implicit transition between Figures A,B,C with

- On event OCOO, {H,M,x,y,z} → {O,M,x,y,z}
- On event OCOF, {O,M,x,y,z} → {H,M,x,y,z}
- On event OCOO, {H,A,x,y,z} → {O,A,x,y,z}
- On event OCOF, {O,A,x,y,z} → {H,A,x,y,z}

Not all transitions are implemented mechanically. For example the ‡ transitions require procedural support.

Figure D1. AFTL states HMI-Manual, HMI-Auto, HOS-Manual, HOS-Auto

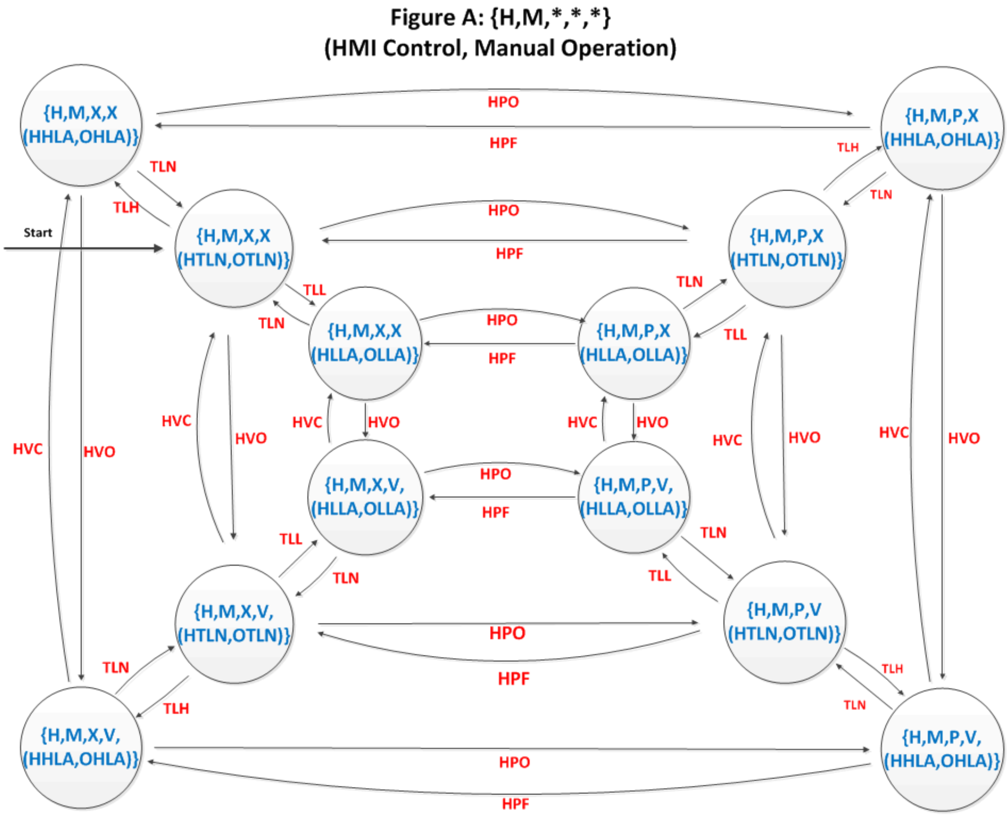


Figure D2. HMI-Manual state diagram.

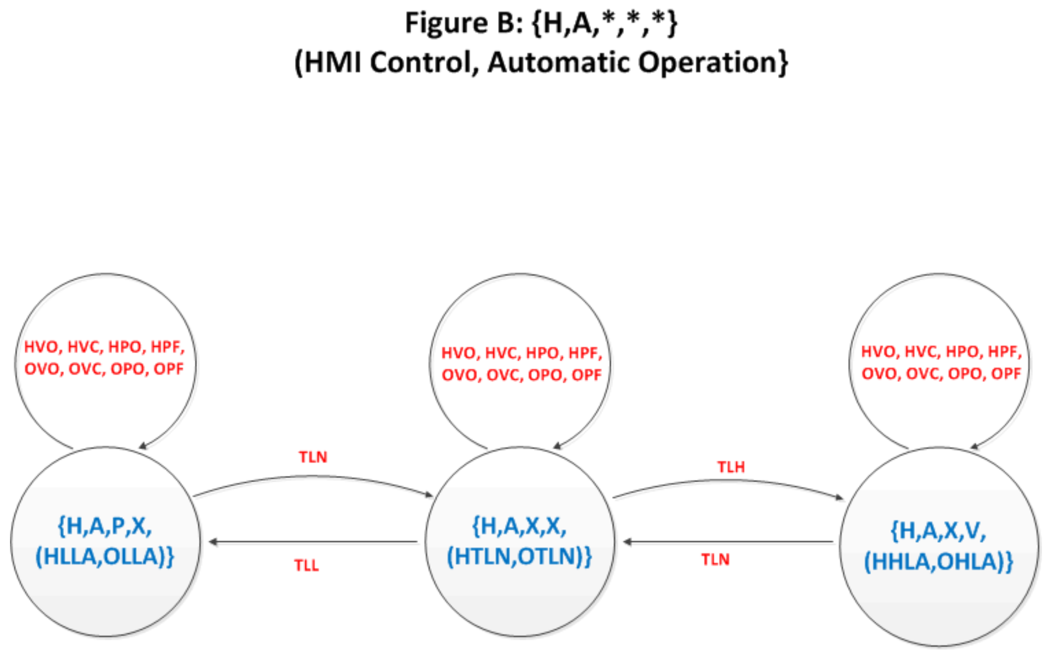


Figure D3. HMI-Auto state diagram.

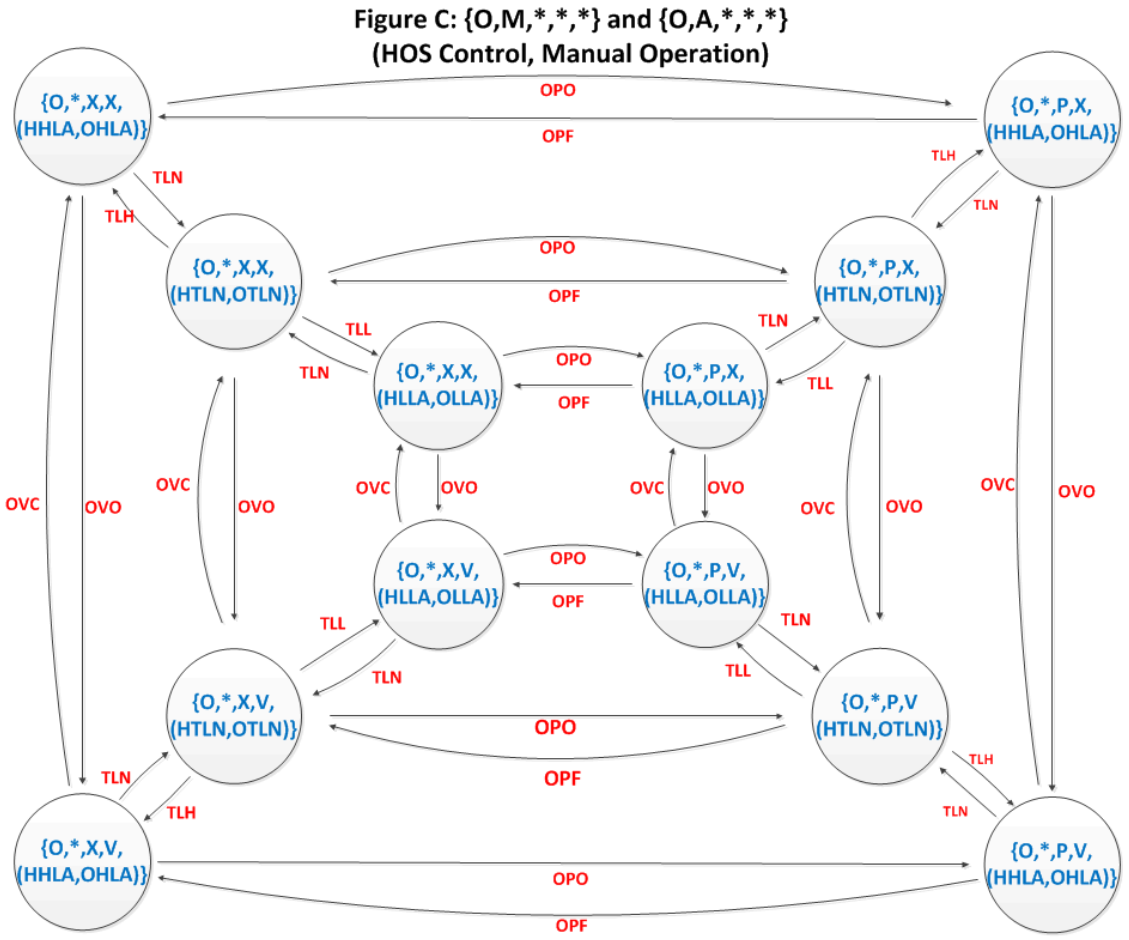
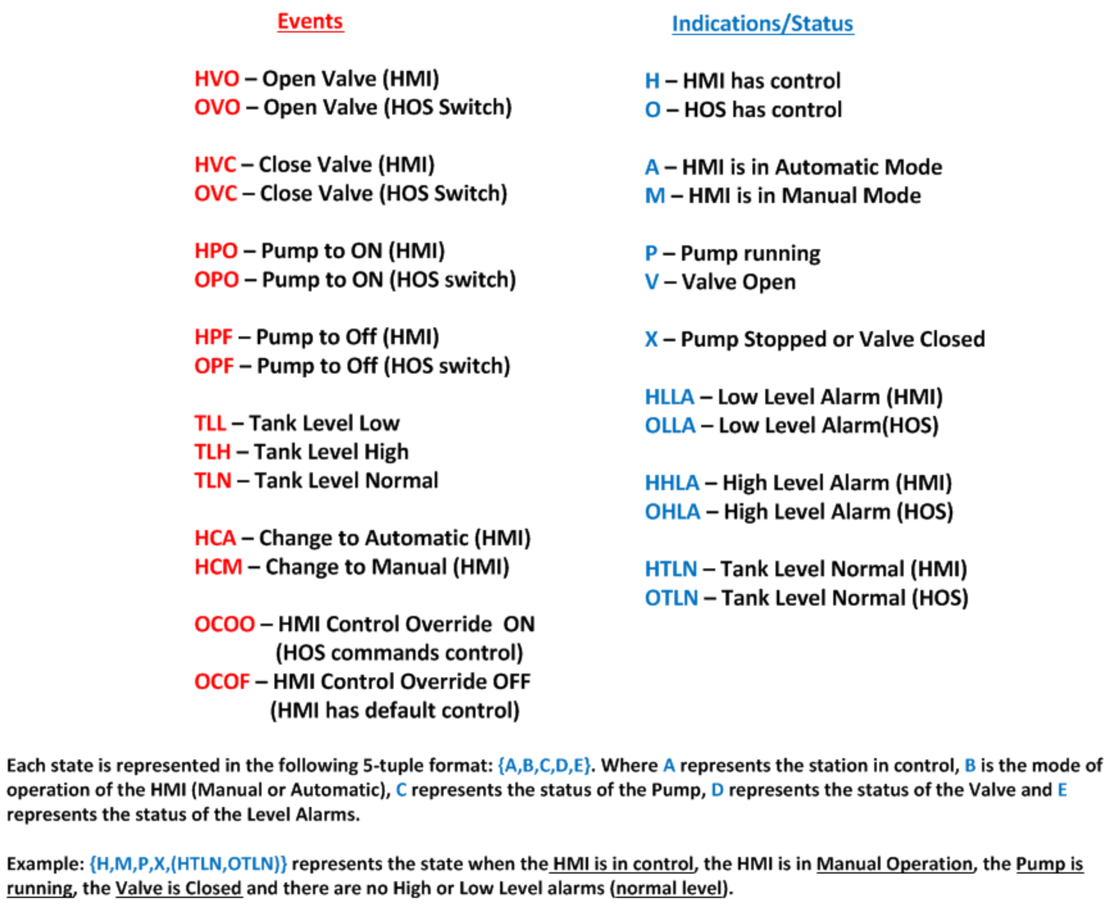


Figure D4. HOS-Manual and HOS-Auto state diagram.

When the HOS takes control and the system is in "Automatic" mode the Manual controls will override any automatic functionality by design. HOS control is designed to be an emergency override where the HMI no longer had control. Once control is returned to the HMI the system will return to "Automatic" mode and any previous pump or valve settings will become effective.



Events

HVO – Open Valve (HMI)
OVO – Open Valve (HOS Switch)

HVC – Close Valve (HMI)
OVC – Close Valve (HOS Switch)

HPO – Pump to ON (HMI)
OPO – Pump to ON (HOS switch)

HPF – Pump to Off (HMI)
OPF – Pump to Off (HOS switch)

TLL – Tank Level Low
TLH – Tank Level High
TLN – Tank Level Normal

HCA – Change to Automatic (HMI)
HCM – Change to Manual (HMI)

OCOO – HMI Control Override ON
(HOS commands control)
OCOF – HMI Control Override OFF
(HMI has default control)

Indications/Status

H – HMI has control
O – HOS has control

A – HMI is in Automatic Mode
M – HMI is in Manual Mode

P – Pump running
V – Valve Open

X – Pump Stopped or Valve Closed

HLLA – Low Level Alarm (HMI)
OLLA – Low Level Alarm(HOS)

HHLA – High Level Alarm (HMI)
OHLA – High Level Alarm (HOS)

HTLN – Tank Level Normal (HMI)
OTLN – Tank Level Normal (HOS)

Each state is represented in the following 5-tuple format: {A,B,C,D,E}. Where A represents the station in control, B is the mode of operation of the HMI (Manual or Automatic), C represents the status of the Pump, D represents the status of the Valve and E represents the status of the Level Alarms.

Example: {H,M,P,X,(HTLN,OTLN)} represents the state when the HMI is in control, the HMI is in Manual Operation, the Pump is running, the Valve is Closed and there are no High or Low Level alarms (normal level).

Figure D5. Legend for state diagrams D1–D4

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. DIGITAL I/O LAB STATE DIAGRAM

This appendix presents state diagrams (Figures E2-E3) for the DIOL. Each state is represented as a 4-tuple {A,B,C,D}, where "A" represents the current status of the proximity sensors as displayed in three bit format "000", "B" is the status of the robot arm moving or stowed, "C" is the number of source proximity sensor during a move and "D" is the number of the destination proximity sensor during a move. The proximity sensors state is encoded from left to right, where a "1" indicates the proximity sensor is active and a "0" indicates the sensor is inactive. Example: "110" indicates sensors 1 and 2 are active while sensor number 3 is inactive. The robot arm has two main states: "STO" and "MA". "STO" indicates that the robot is in the stored position (the ready state of the robot arm) and has not been given a command to move. "MA" is when the robot arm is moving. Lastly, the last two numbers of the tuple represent the source and destination sensor for an object movement request. For example, (100,MA,1,3) represents when the first proximity sensor is active, the robot arm is moving and there is a request to move the object at sensor "1" (source) to sensor "3" (destination).

The transitions between states are. "S*xy*", where *x* represents the sensor number and *y* represents the change in status of the sensor ("A" for activated or "I" for inactivated). For example, CMO*x*-*y* is the command to move an object from sensor *x* to sensor *y*; as another; MC indicates that the move is complete and the robot arm is returning to the "STO" position.

The two common error states are labeled "E" and "E3". "E" represents the error that occurs when an object fails to arrive at the destination or is knocked off of the source by the robot arm. "E3" represents the condition when all 3 sensors are active and there are no open sensors to which the object can move. It is assumed in this diagram that the change in status of the proximity sensors (when a command is issued) is caused by the robot arm and not by a user moving the objects at the same time. Once the robot has stopped moving, it is assumed the user will change the configuration and move the objects, causing sensors to change state. Figure E1 shows a normal sequence of states and transitions when commanding the robot arm to move an object from sensor 1 to sensor 3.

The following steps demonstrate 2 separate possible scenarios when traversing the DIOL state diagram and are meant as an example of how the diagram is used. Referring to Figure E3, as an example, the following steps show the sequence of states and transitions experienced in the normal operation of the DIOL:

1) The “Ready State” (the start state of the DIOL) is the blue colored state labeled “0”. In the “Ready State” all 3 of the proximity sensors are inactive, the robot arm is in the “STO” position and there is not a command to move any objects.
2) An object is placed on proximity sensor number 1 resulting in the transition from state “0” to state “4”.
3) The order to move the object from sensor number 1 to sensor number 3 is given “CMO1-3” causes the transition from state “4” to state “100,MA,1,3”.
4) The robot arm moves to sensor position 1 and picks up the object causing sensor 1 to be inactive resulting in transition “S1I” to state “000,MA,1,3”.
5) The robot arm places the object on sensor number 3 causing sensor 3 to become active and cause the transition “S3A” to state “001,MA,1,3”.
6) The robot returns to the “STO” position and the object move command is completed causing a transition on “MC” to state number “1” where the robot arm is ready for a new command to move an object.

Referring to Figure E2, as an example, the following steps demonstrate the sequence of transitions and states in order to arrive at an error state:

1) From state “0” the “Ready State” an object is placed on the first sensor causing transition “S1A” to state “4”.
2) Another object is placed on sensor number 3 causing a transition on “S3A” to state “5”
3) From state “5” another object is placed in sensor 2 causing a transition on “S2A” to the error state “E3” indicating that all the sensors are full and an object needs to be removed in order to continue. (see Figures E2-E3)

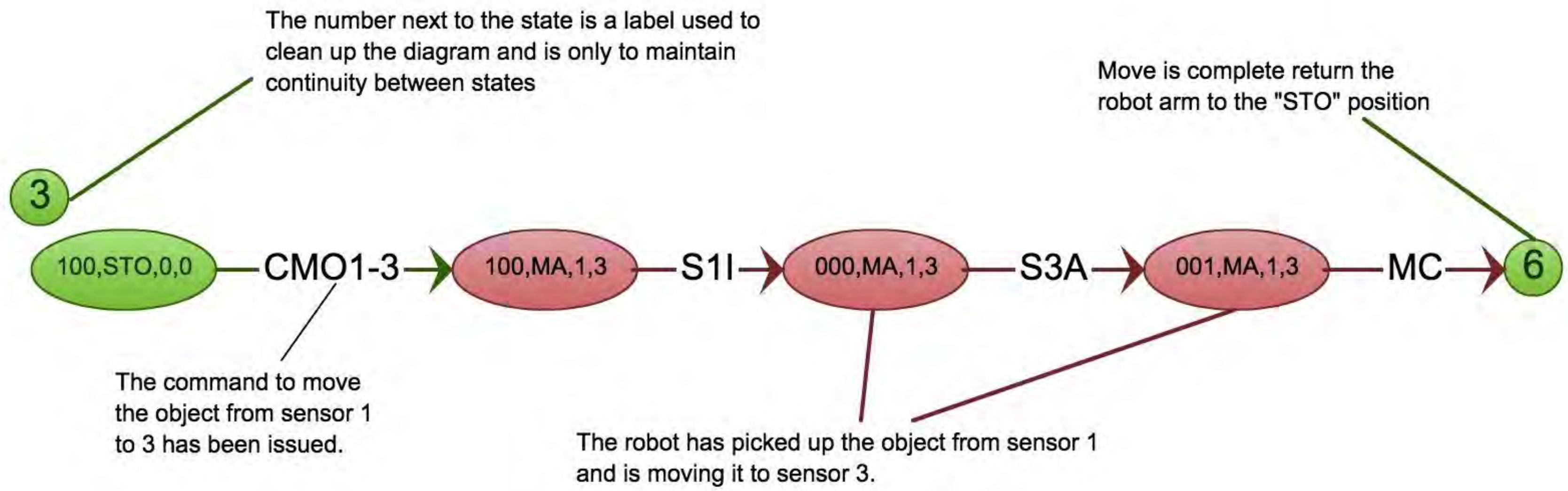
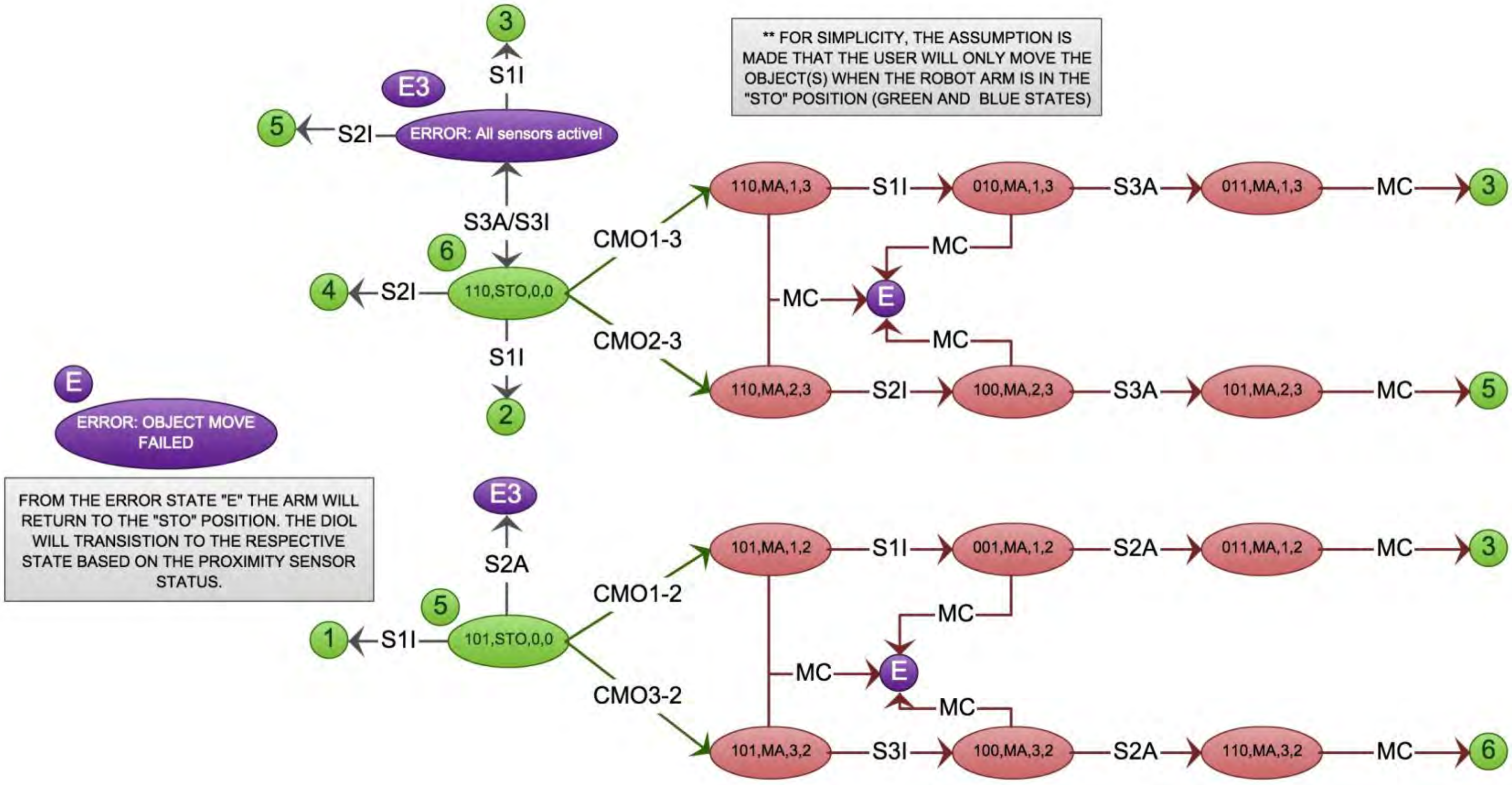

The number next to the state is a label used to clean up the diagram and is only to maintain continuity between states
Move is complete return the robot arm to the "STO" position
3
100,STO,0,0
CMO1-3
100,MA,1,3
S1I
000,MA,1,3
S3A
001,MA,1,3
MC
6
The command to move the object from sensor 1 to 3 has been issued.
The robot has picked up the object from sensor 1 and is moving it to sensor 3.

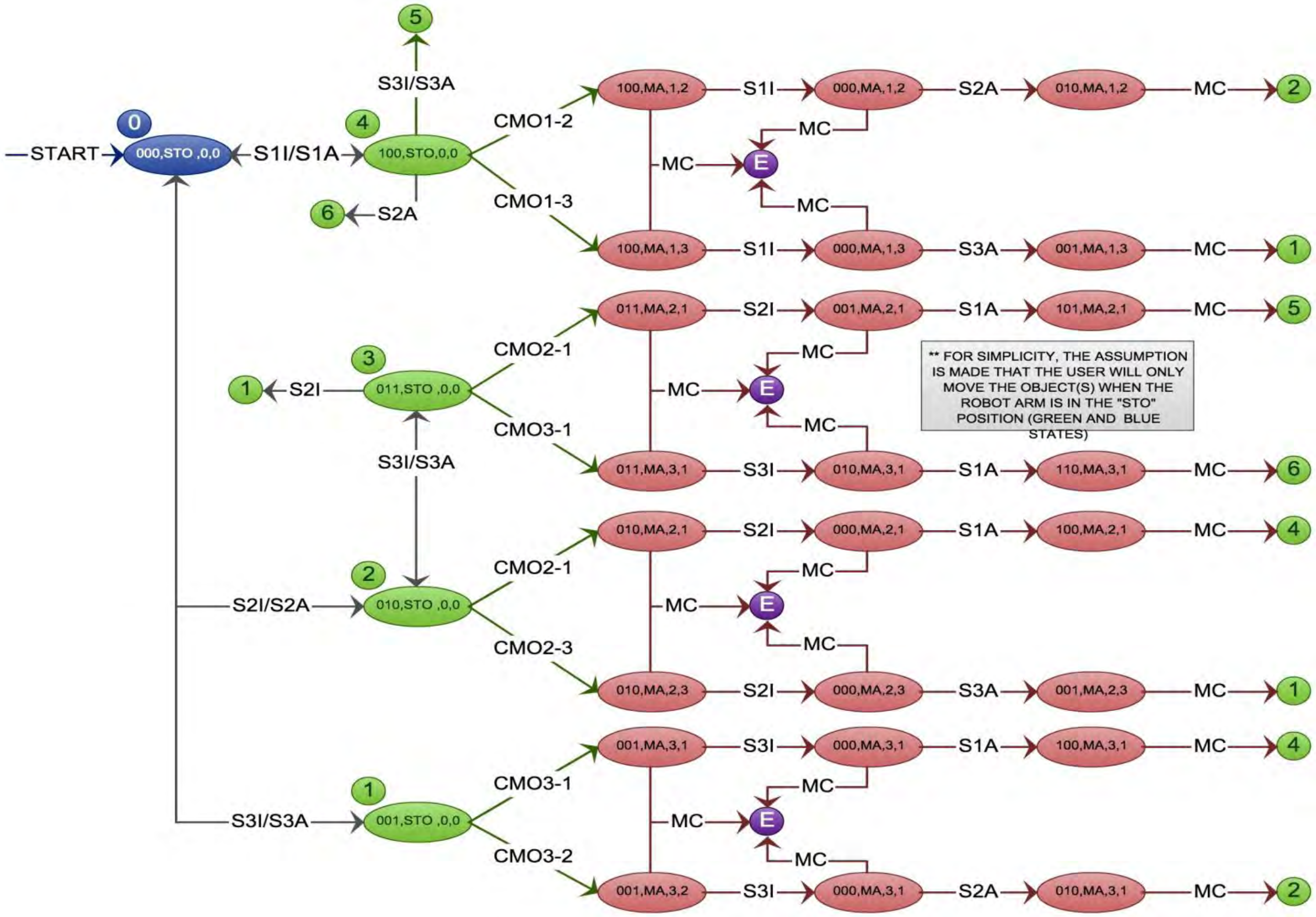
Figure E1. An example state diagram indicating a normal movement of an object from sensor 1 to sensor 3.



** FOR SIMPLICITY, THE ASSUMPTION IS MADE THAT THE USER WILL ONLY MOVE THE OBJECT(S) WHEN THE ROBOT ARM IS IN THE "STO" POSITION (GREEN AND BLUE STATES)

FROM THE ERROR STATE "E" THE ARM WILL RETURN TO THE "STO" POSITION. THE DIOL WILL TRANSISTION TO THE RESPECTIVE STATE BASED ON THE PROXIMITY SENSOR STATUS.

Figure E2. DIOL state diagram part 1of 2



** FOR SIMPLICITY, THE ASSUMPTION IS MADE THAT THE USER WILL ONLY MOVE THE OBJECT(S) WHEN THE ROBOT ARM IS IN THE "STO" POSITION (GREEN AND BLUE STATES)

Figure E3. DIOL state diagram part 2 of

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F TESTING PROCEDURES

TABLE OF CONTENTS

The following procedures are written to exercise the design requirements as outlined in Chapter III. The testing procedure follows the test plan from Chapter VI

F1. HARDWARE CONNECTION SETUP

The initial setup should be followed as a familiarization and overview of the system or if there were any changes to the testbed. Additionally, the testbed was designed with portability in mind and as such may have to be initialized periodically after being moved or after being used extensively for research. The following procedure ensures that all the hardware is installed and aligned properly to ensure a solid baseline from which to test.

Use the Table F1 and Figures F1-F10 to properly connect all cables needed to establish the proper hardware connections needed to operate the testbed. In Table F1 a unique number is given to each cable in order to identify how and where each end connection should be placed. Each figure will reference one or more cable numbers in Table F1.

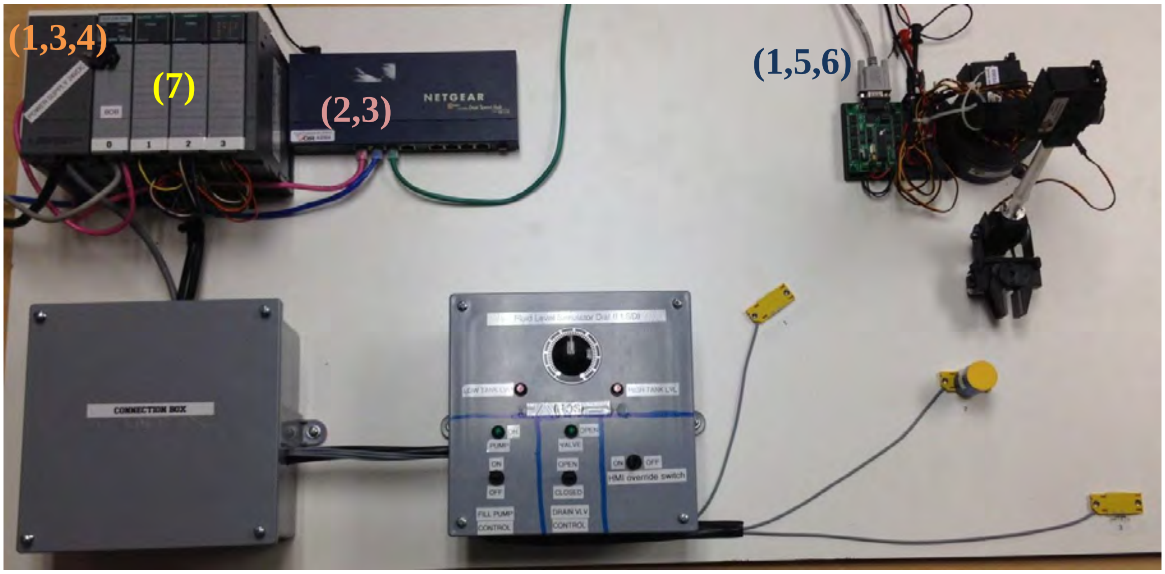


Figure F1. MCS Testbed overview

Note: not shown above: 9VDC power supply (5), power strip (4,6) and engineering workstation (2)

Table F1. Cables needed to initialize the testbed (w/ source and destination)

Cable #	***Description***	***Source connection***	***Destination connection***
1	Serial (RS-232) Connection	PLC CPU Module 0	Robot Arm (SSC-32)
2	Ethernet connection (blue)	HMI/ Eng. Workstation (Alice)	Ethernet Hub
3	Ethernet connection (pink)	PLC CPU Module 0	Ethernet Hub
4	Power connection (PLC)	Power strip (plug)	PLC Rack Power supply (24VDC)
5	9V DC Power supply (Robot Arm)	9VDC power supply	Robot Arm (SSC-32) Battery terminal
6	SSC-32 Circuit Board power	Power strip (transformer)	Robot Arm (SSC-32) round plug
7	Electrical Wiring (connection box to PLC Modules 1-3)	PLC Modules 1-3	HOS and Proximity sensors (via Conn. Box)

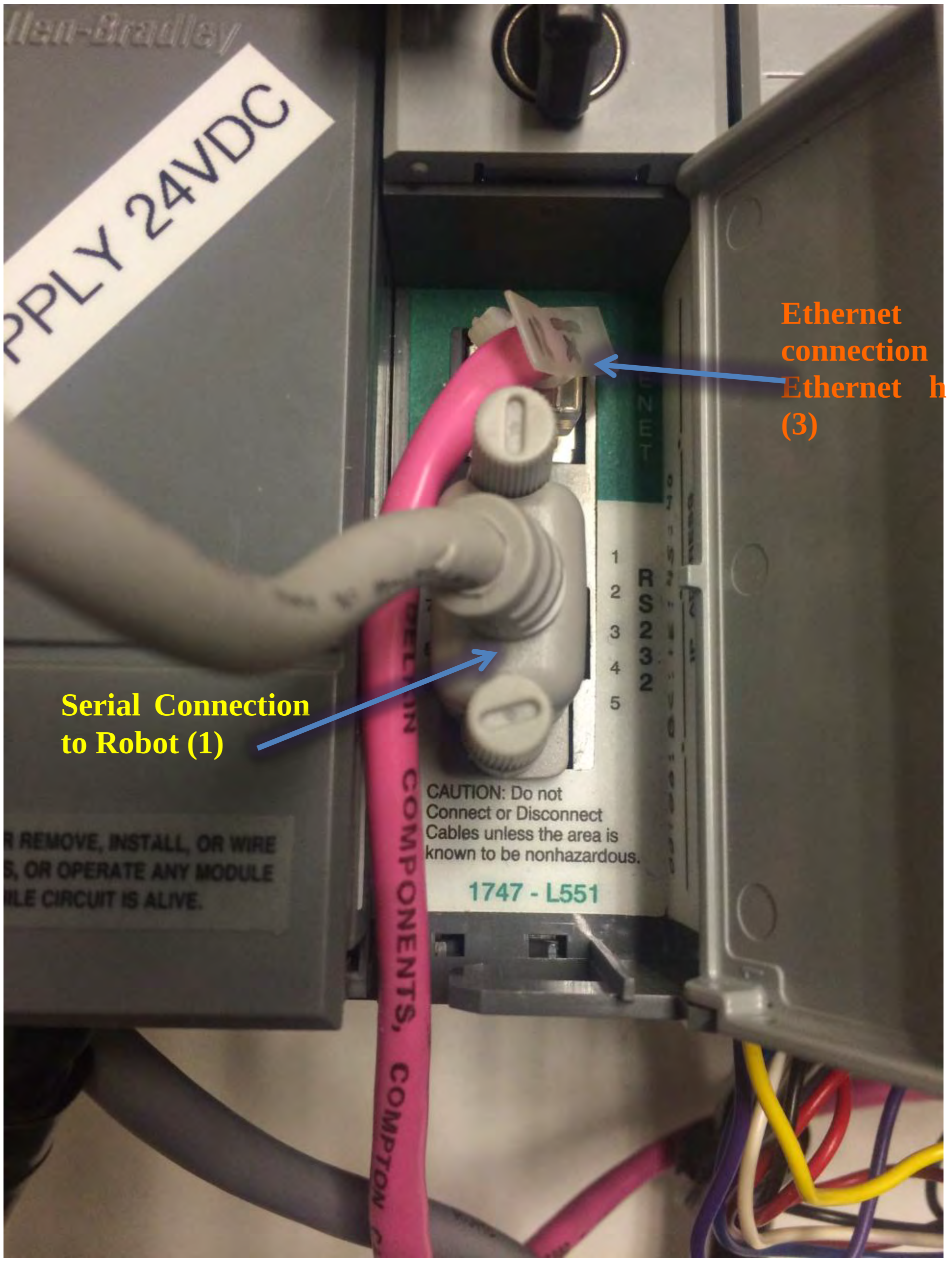

Allen-Bradley
PPLY 24VDC
Ethernet connection to Ethernet hub (3)
Serial Connection to Robot (1)
1
2
3
4
5
RS232
CAUTION: Do not Connect or Disconnect Cables unless the area is known to be nonhazardous.
1747 - L551
REMOVE, INSTALL, OR WIRE
S, OR OPERATE ANY MODULE
ILE CIRCUIT IS ALIVE.
COMPONENTS, COMPTON

Figure F2. PLC CPU Module 0

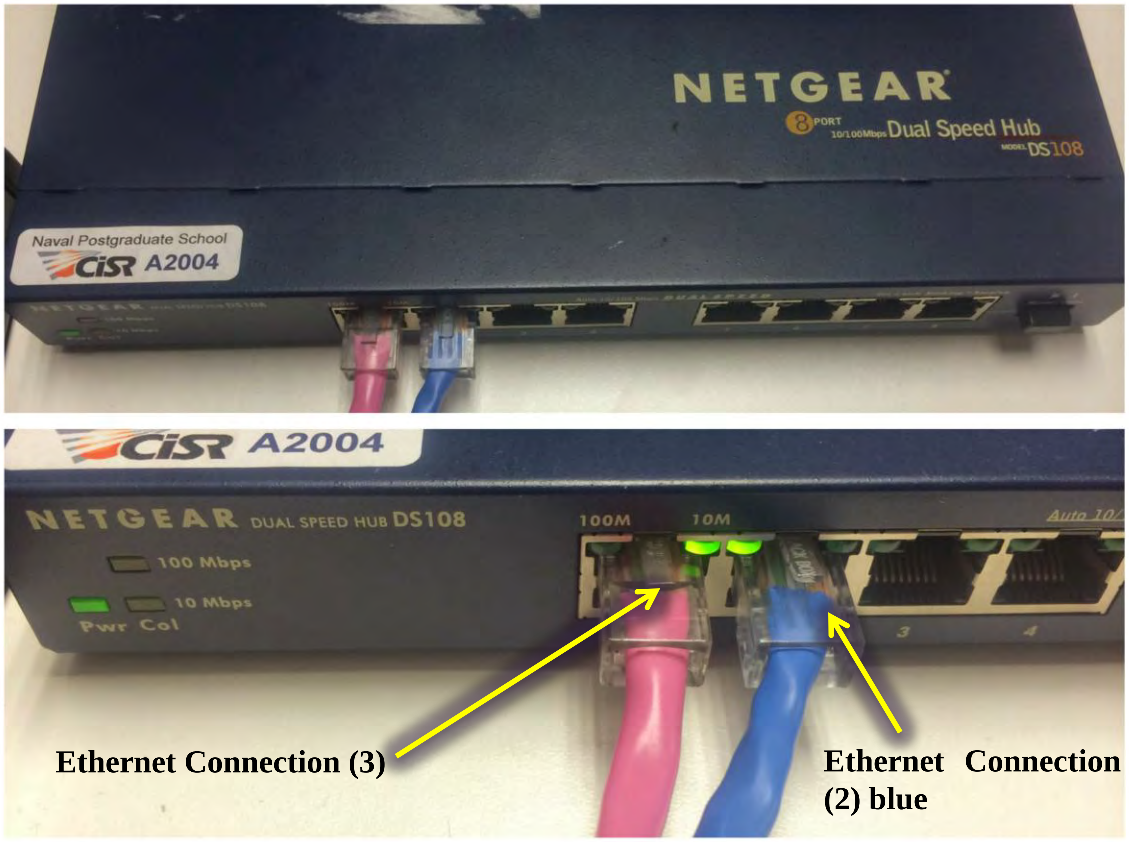


Figure F3. Ethernet Hub Connections

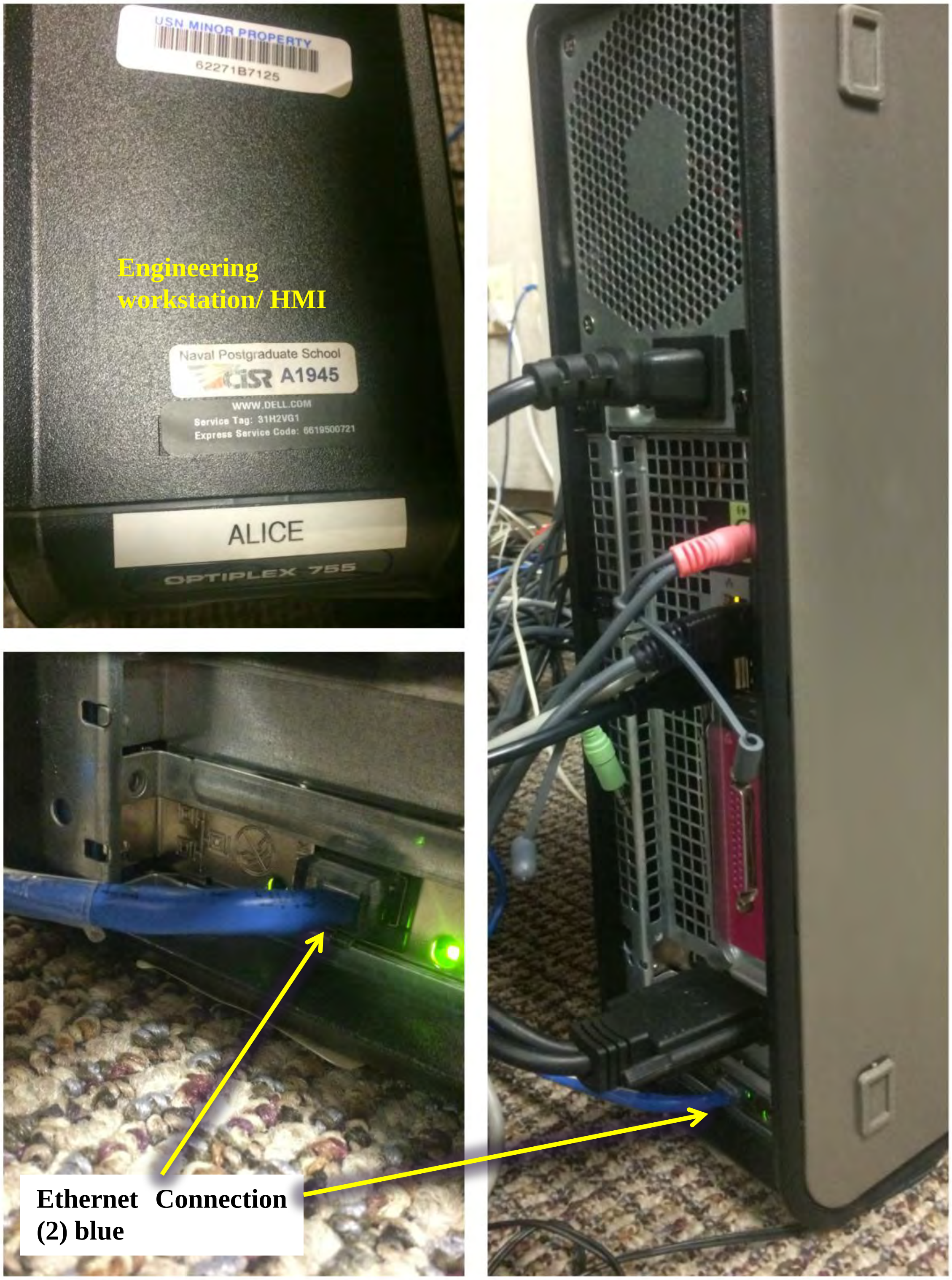

USN MINOR PROPERTY
62271B7125
Engineering workstation/ HMI
Naval Postgraduate School
CISR A1945
WWW.DELL.COM
Service Tag: 31H2VG1
Express Service Code: 6619500721
ALICE
OPTIPLEX 755
Ethernet Connection (2) blue

Figure F3. Engineering workstation and HMI

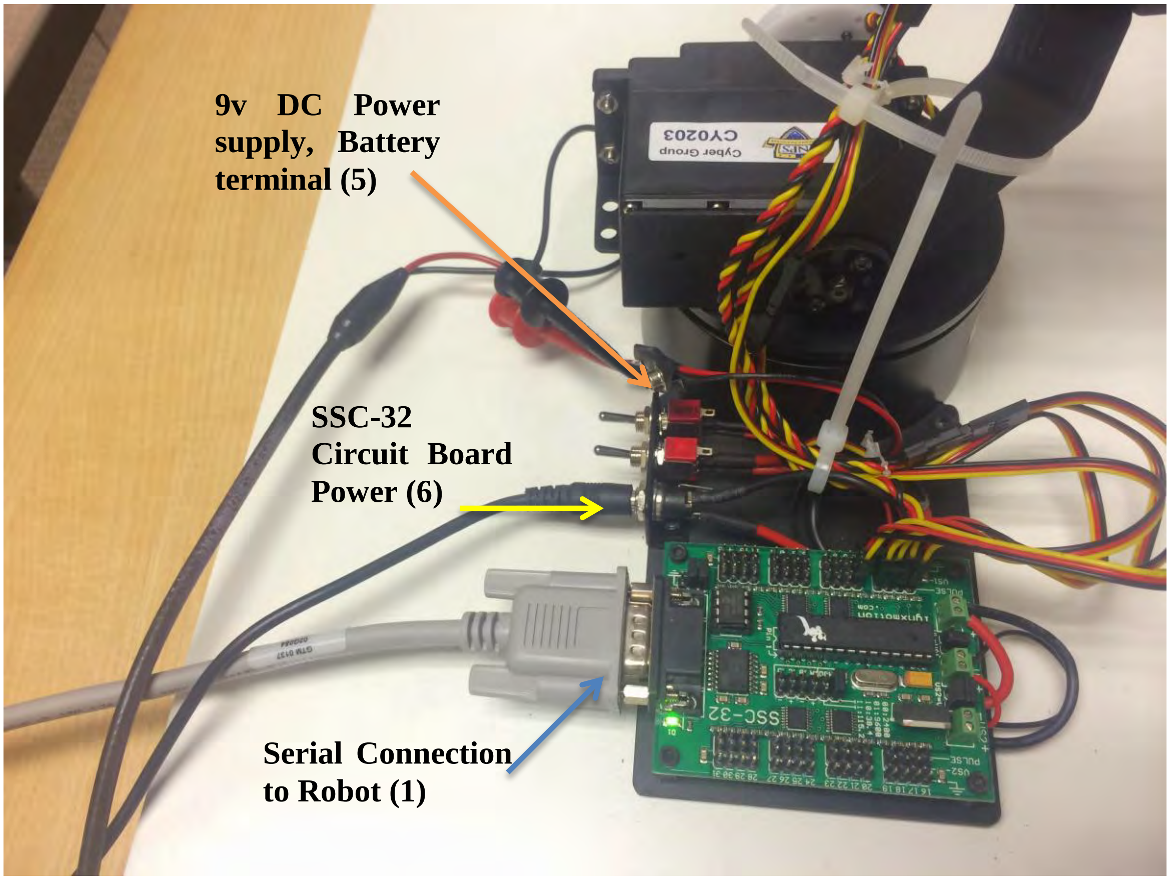

9v DC Power supply, Battery terminal (5)
SSC-32 Circuit Board Power (6)
Serial Connection to Robot (1)
CY0203
Cyber Group
SSC-32

Figure F4. Robot Arm SSC-32 connections

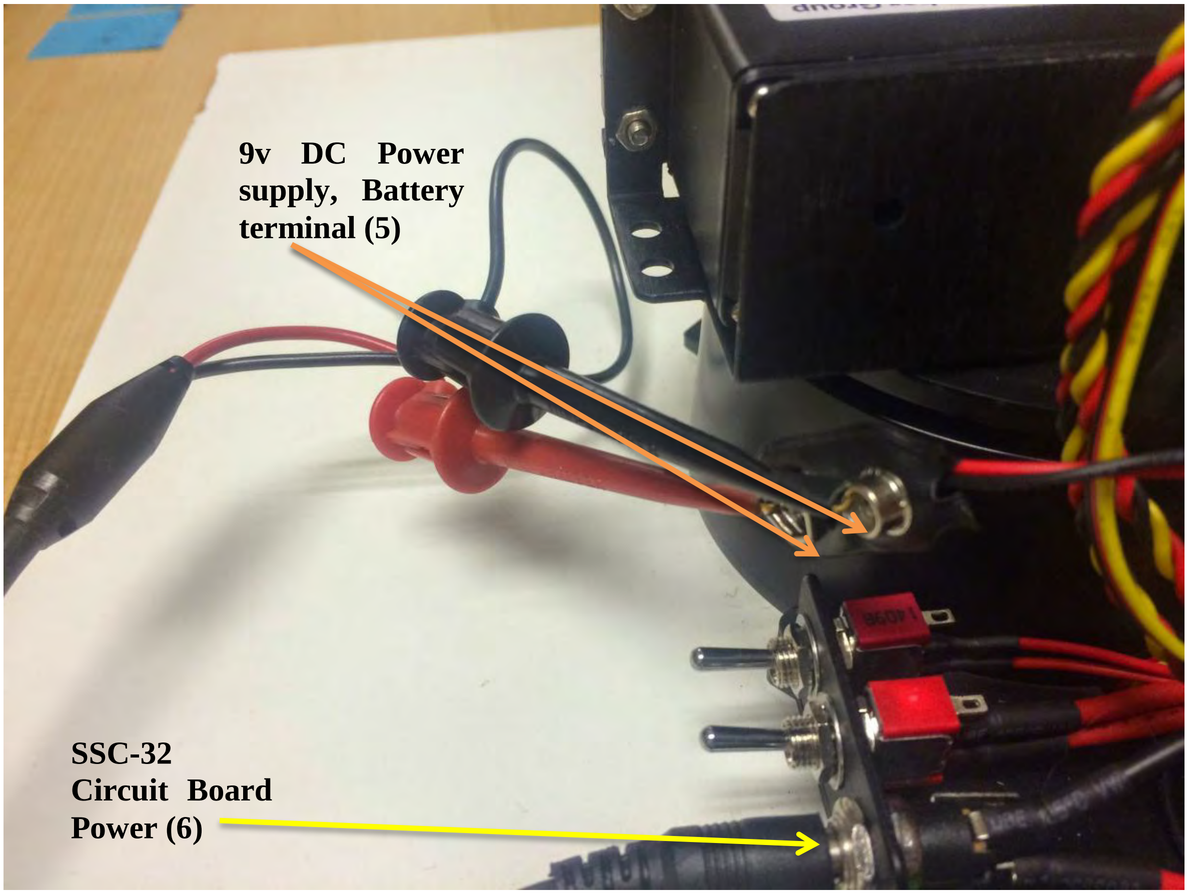
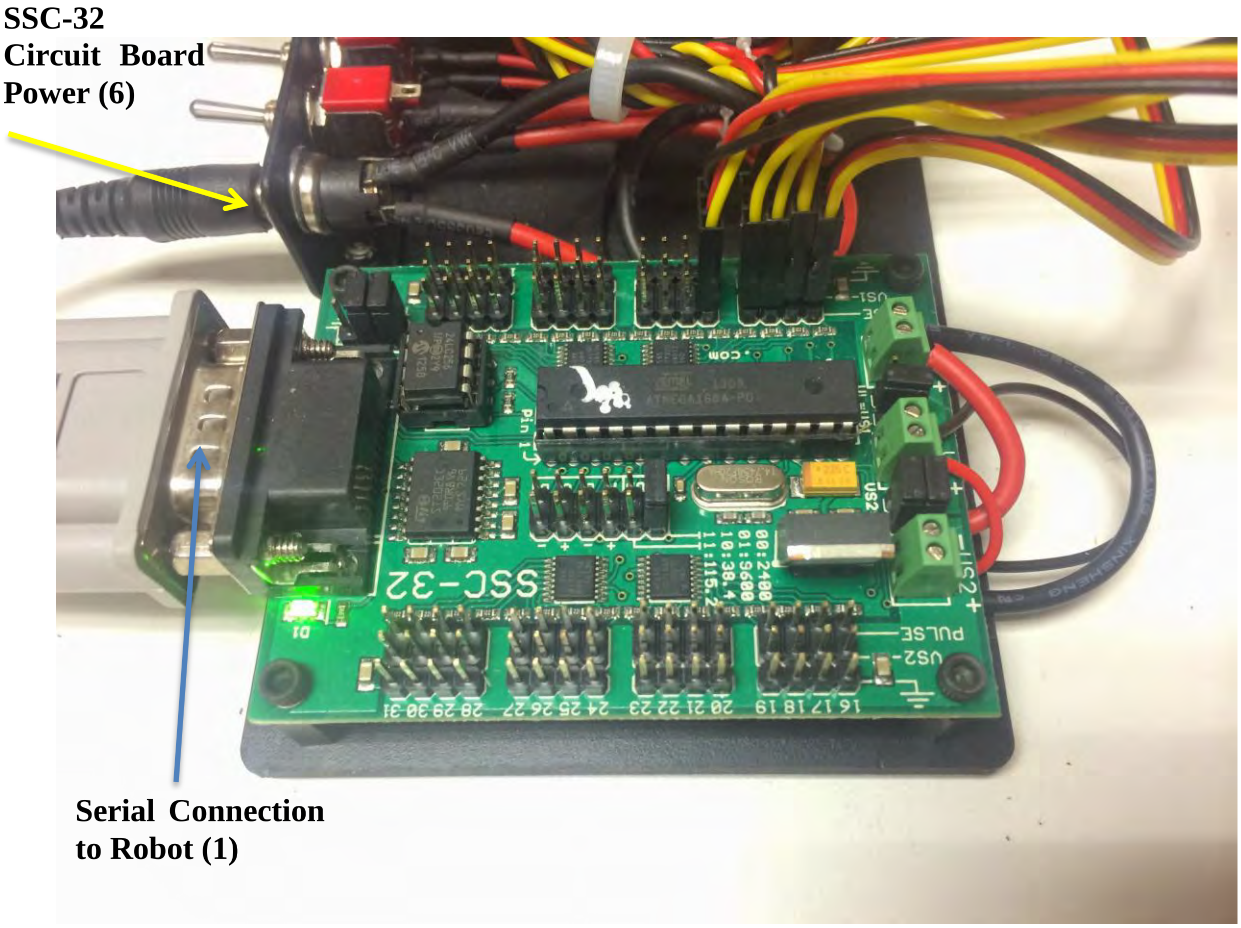

9v DC Power supply, Battery terminal (5)
SSC-32 Circuit Board Power (6)

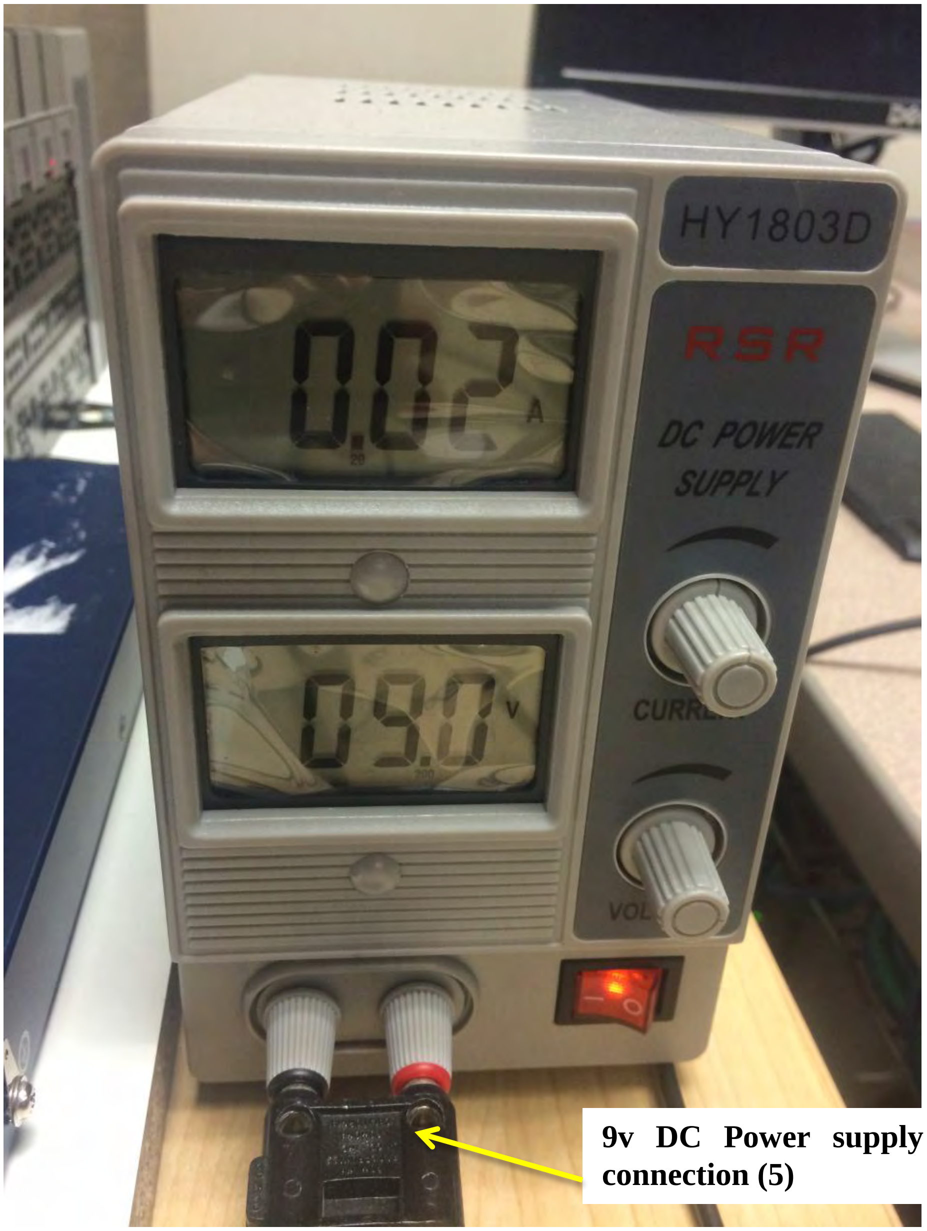
Figure F5. Robot arm (SSC-32) Battery Terminal, Switches and Circuit board power



SSC-32 Circuit Board Power (6)

Serial Connection to Robot (1)

Figure F6. SSC-32 Servo Controller Close-up



9v DC Power supply connection (5)

Figure F7. 9VDC Power supply close-up

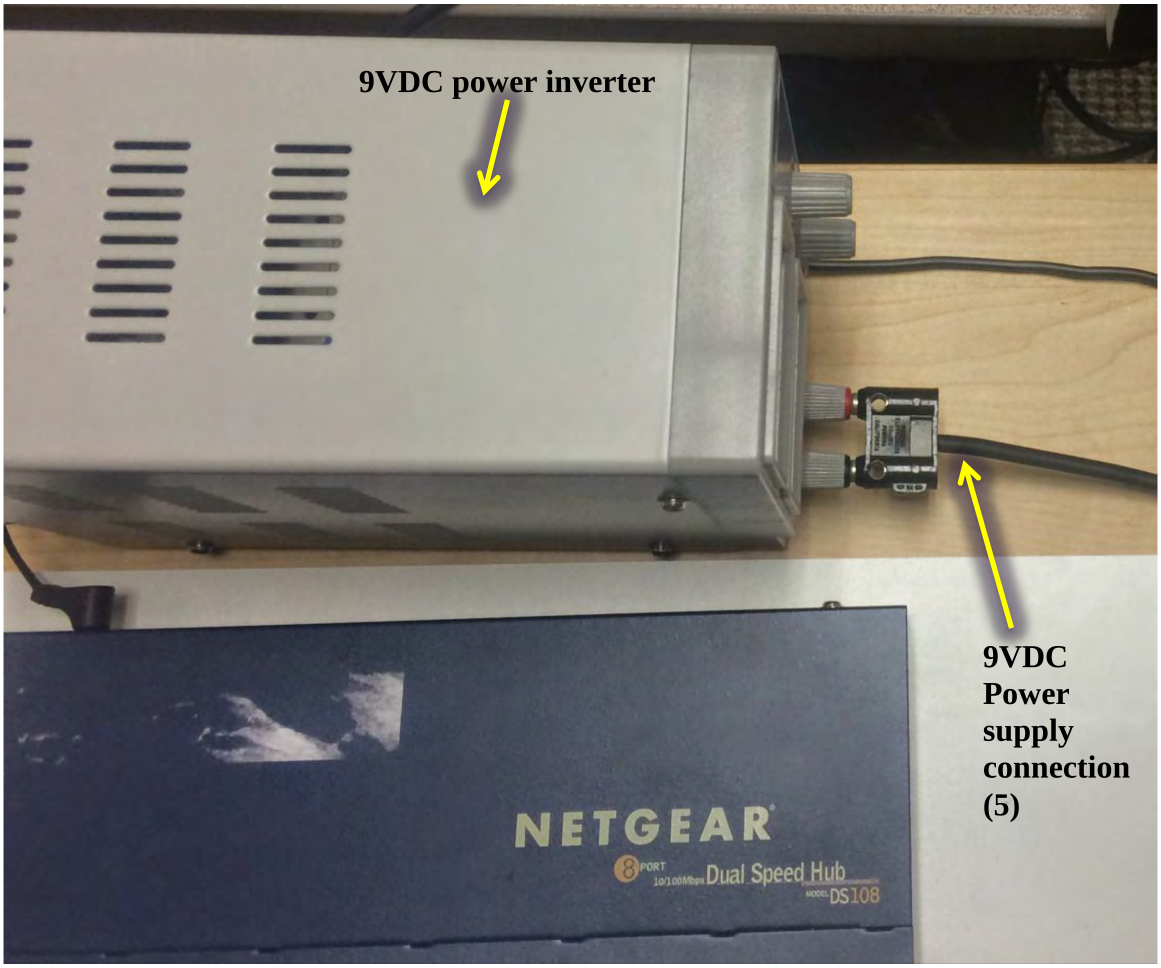

9VDC power inverter
9VDC Power supply connection (5)
NETGEAR
8 PORT 10/100Mbps Dual Speed Hub
MODEL DS108

Figure F8. 9VDC power inverter

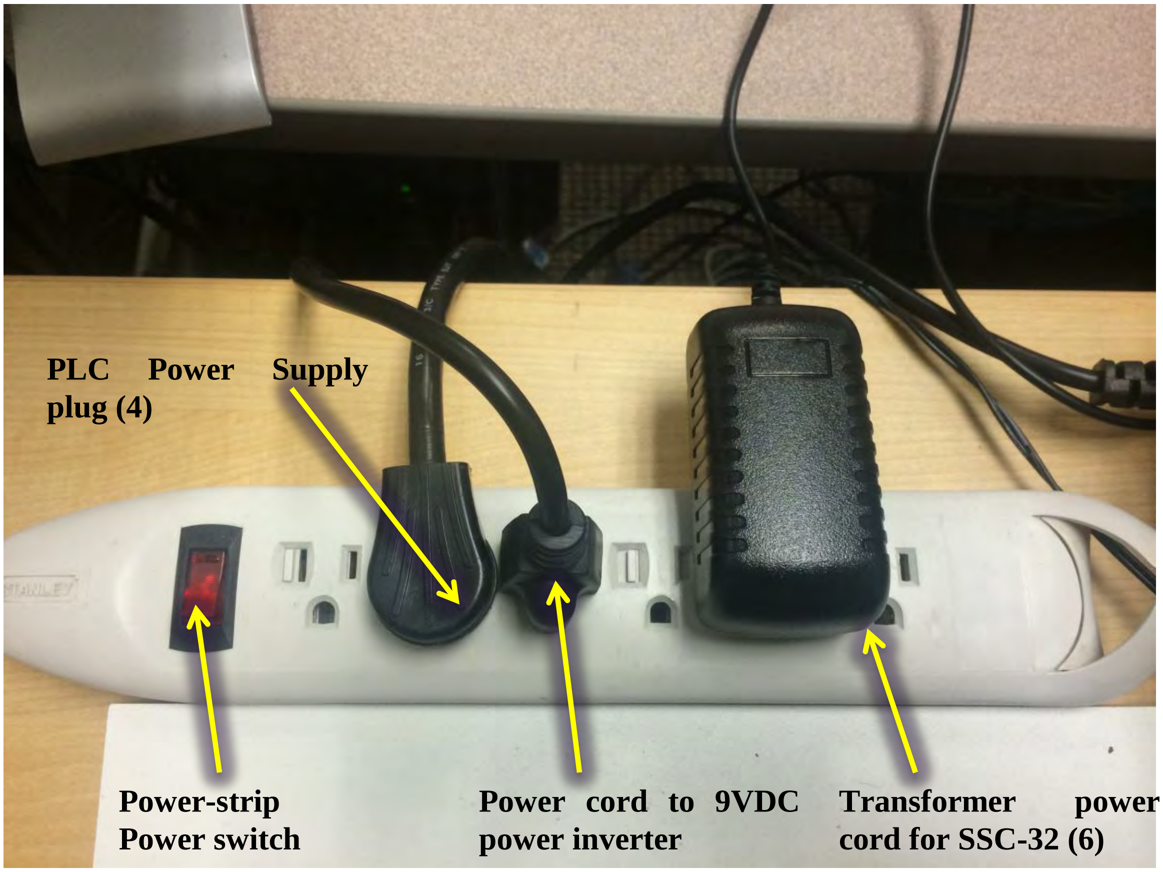
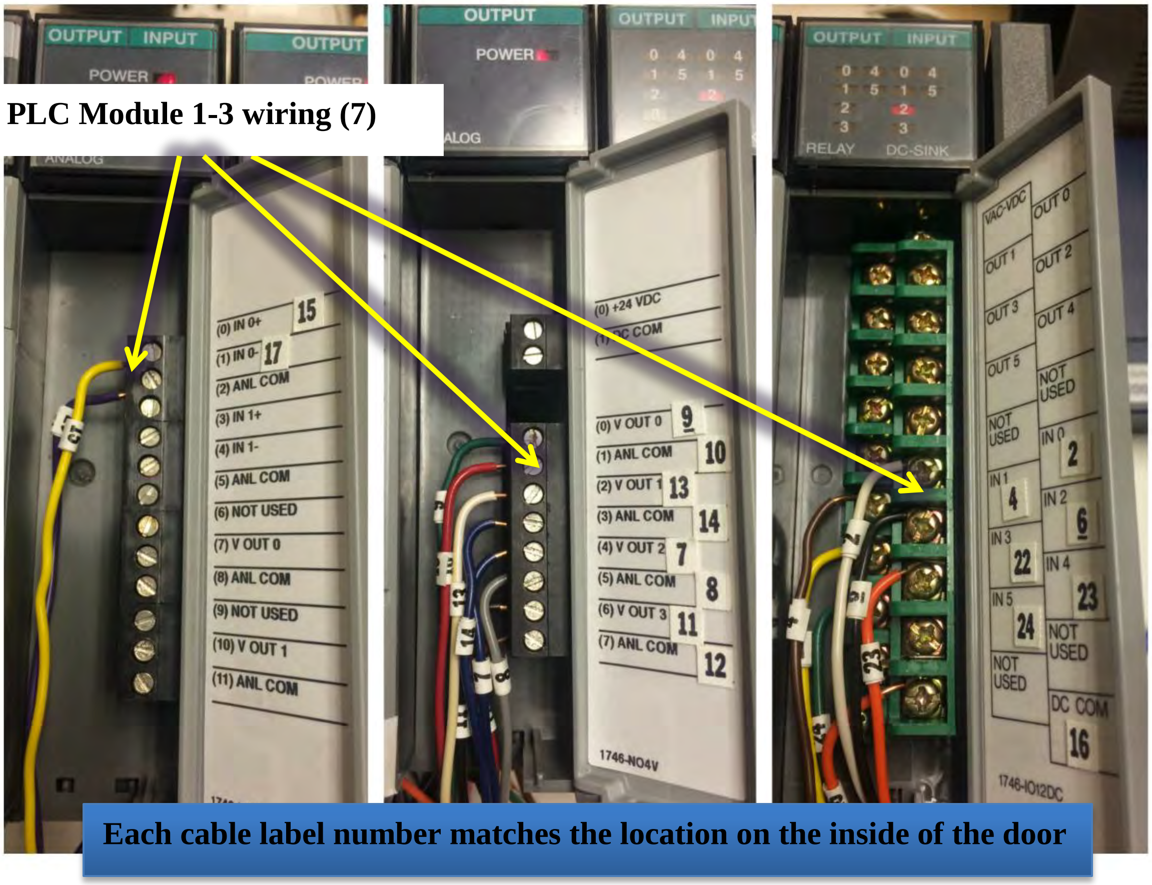

PLC Power Supply plug (4)
Power-strip Power switch
Power cord to 9VDC power inverter
Transformer power cord for SSC-32 (6)

Figure F9. Power strip with plugs


PLC Module 1-3 wiring (7)

Each cable label number matches the location on the inside of the door

Figure F10. PLC I/O Modules 1-3 and associated wires (7)

(See Appendix A and B for an AFTL wiring schematic and diagram)

F2. HARDWARE CONFIGURATION

Once all hardware is wired up and connected properly, as described in chapter F1, the next step is to verify the switch positions on the PLC and the SSC-32 robot arm servo controller, 9VDC power supply and power strip. The following pictures show the required settings for the PLC, SSC-32 servo controller (robot arm), the variable DC power supply and power strip.

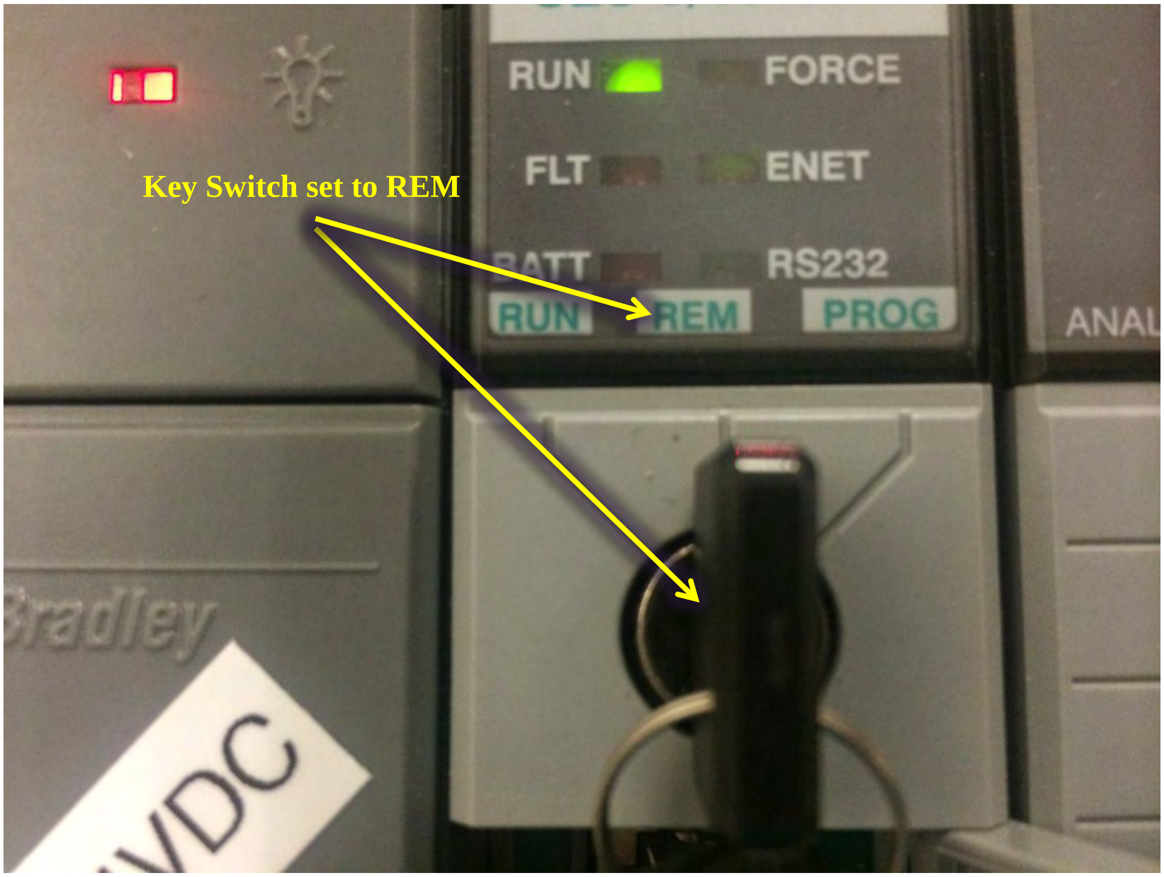


Figure F11. PLC CPU (module 0) key switch setting (“REM”)

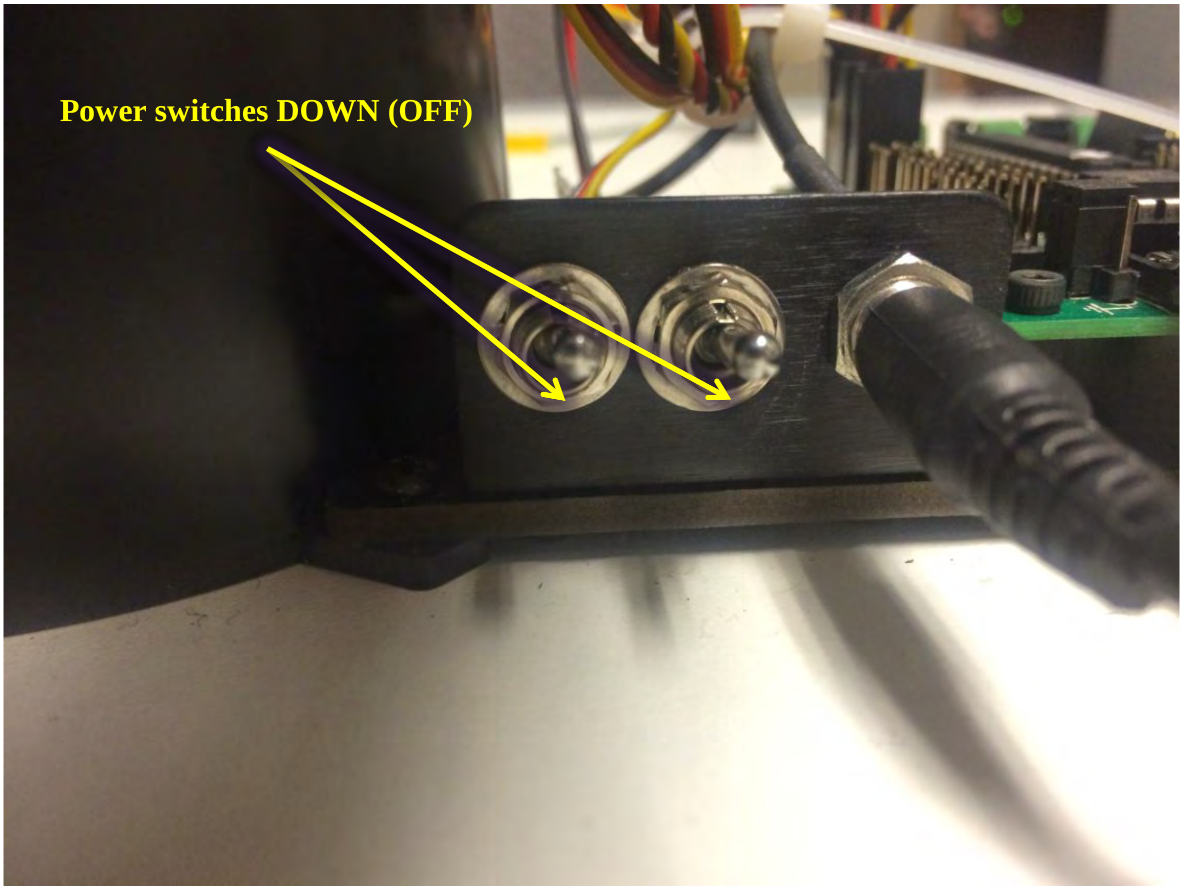


Figure F12. SSC-32 Servo controller (robot arm) power switches set Down (OFF)

Important: Before power can be applied to the SSC-32 servo controller (robot arm) the power supply must be checked to ensure it is producing the correct voltage (9VDC). Ensure the SSC-32 power switches are OFF before proceeding.

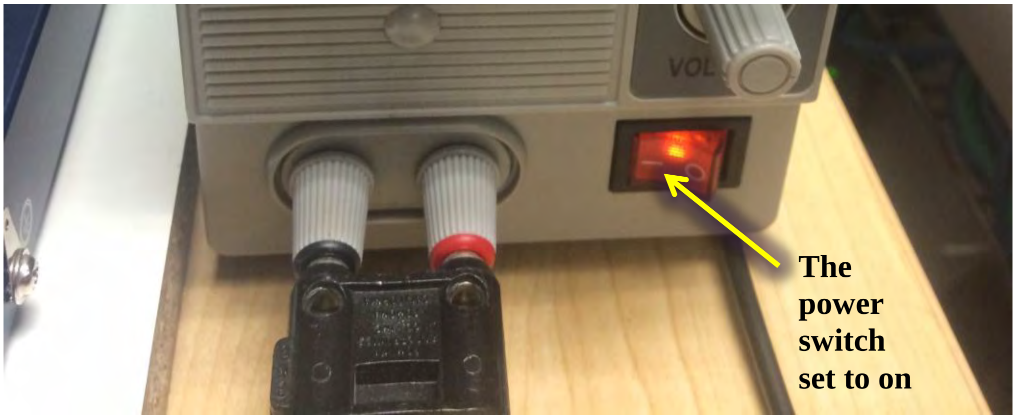


Figure F13a. Variable DC power supply powered ON only

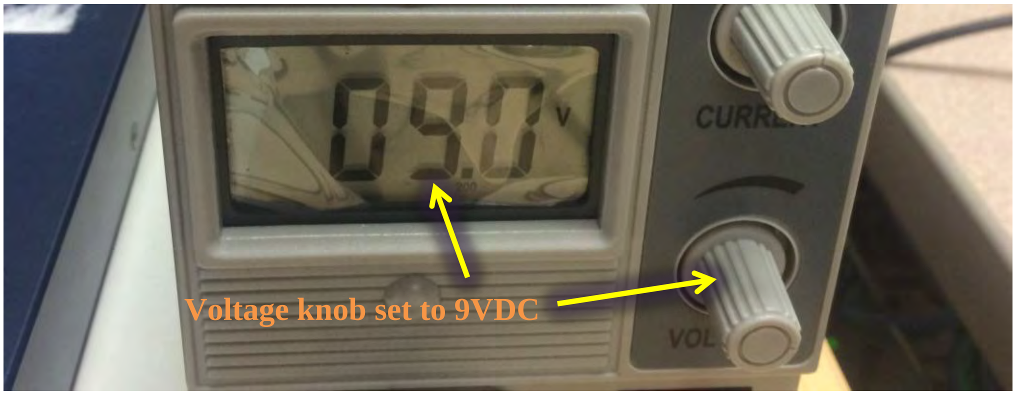


Figure F13b. Adjust the power supply to indicate 9VDC

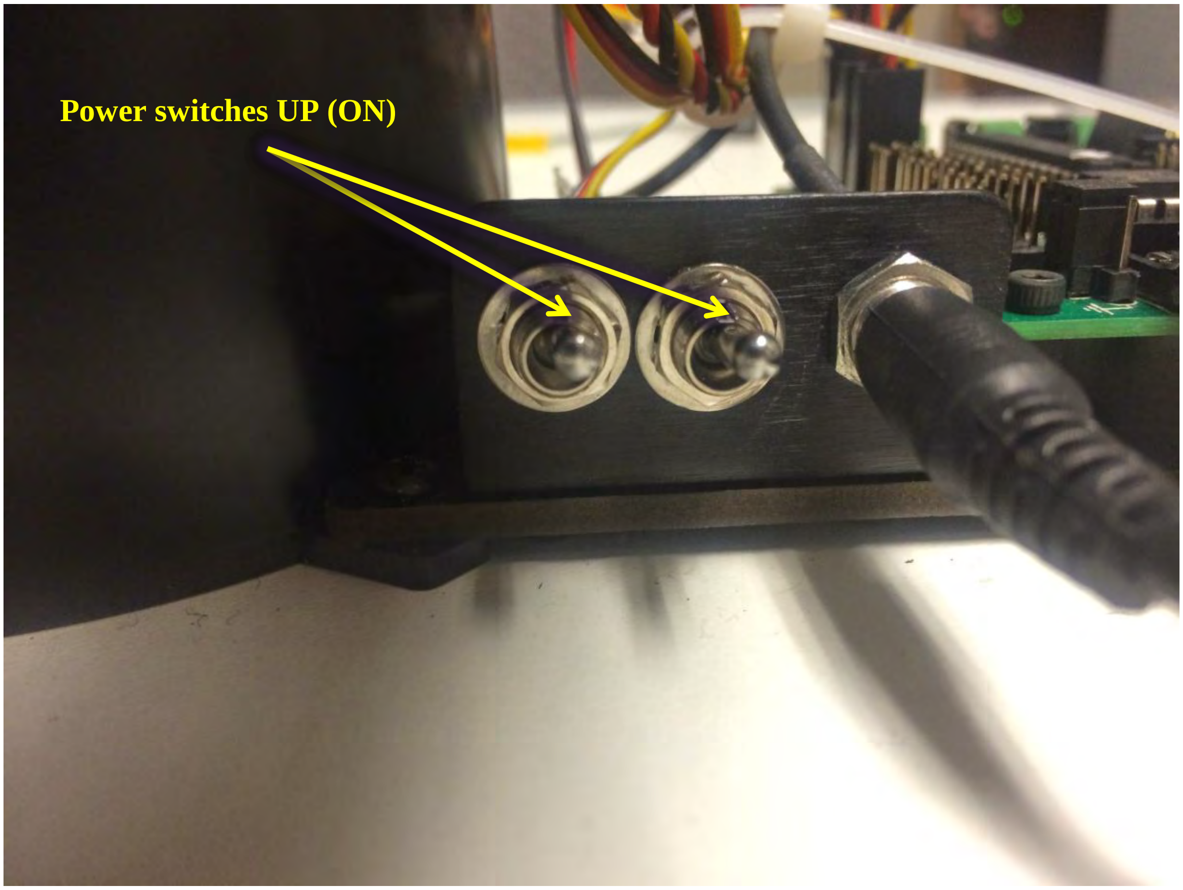


Figure F12. SSC-32 Servo controller (robot arm) power switches set to UP (ON)

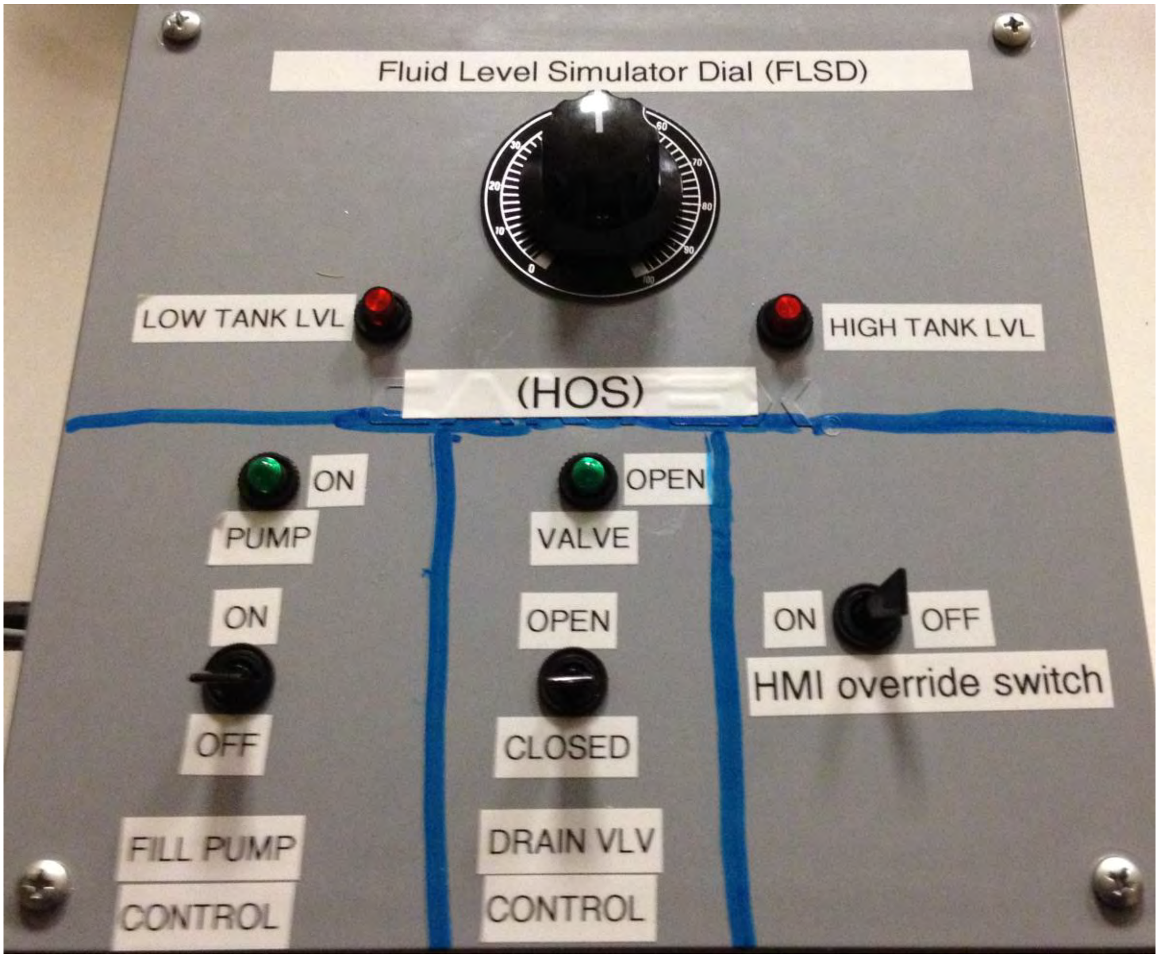


Figure F14. Set all the HOS switches to "OFF/CLOSED" and the FLSD to 50%

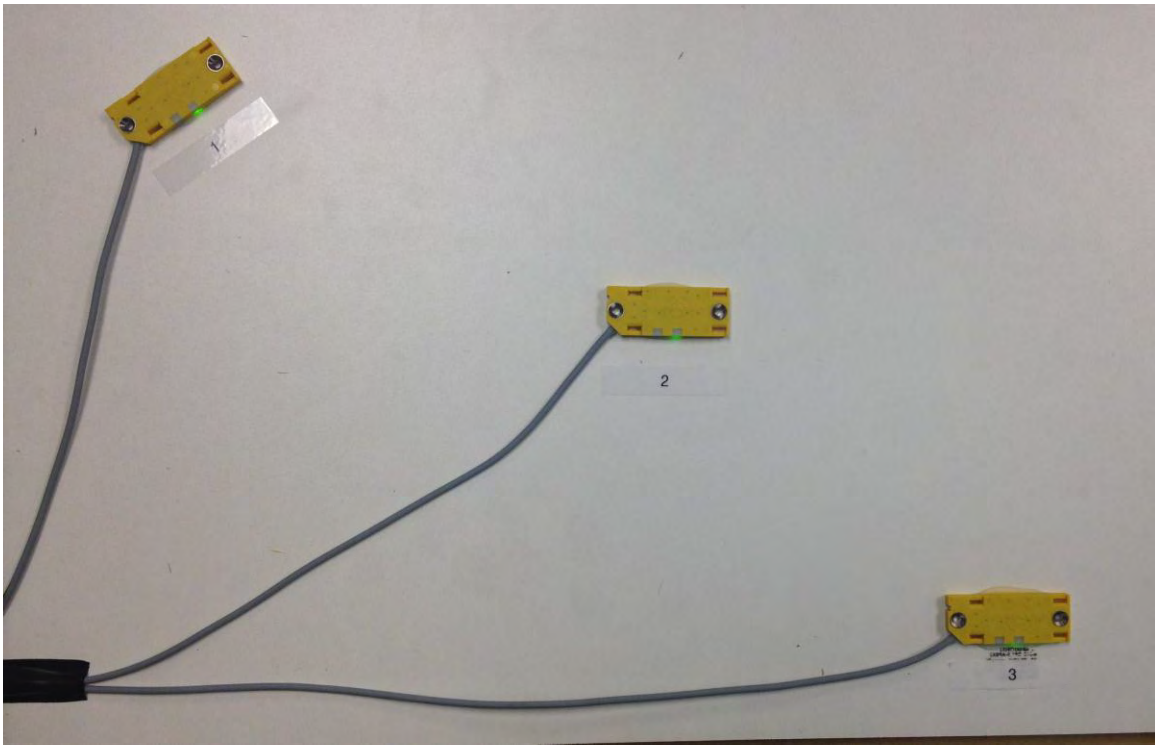


Figure F15. Ensure there are NO objects resting on the proximity sensors [25]

After verifying all the connections and switch settings above are as indicated or described, turn the power switch on the power strip to the ON position (see Figure F9).

Verify that the PLC performs the following sequence as it is powered up. The PLC should remain in the state indicated by number 3 “Normal” in Figure F16.

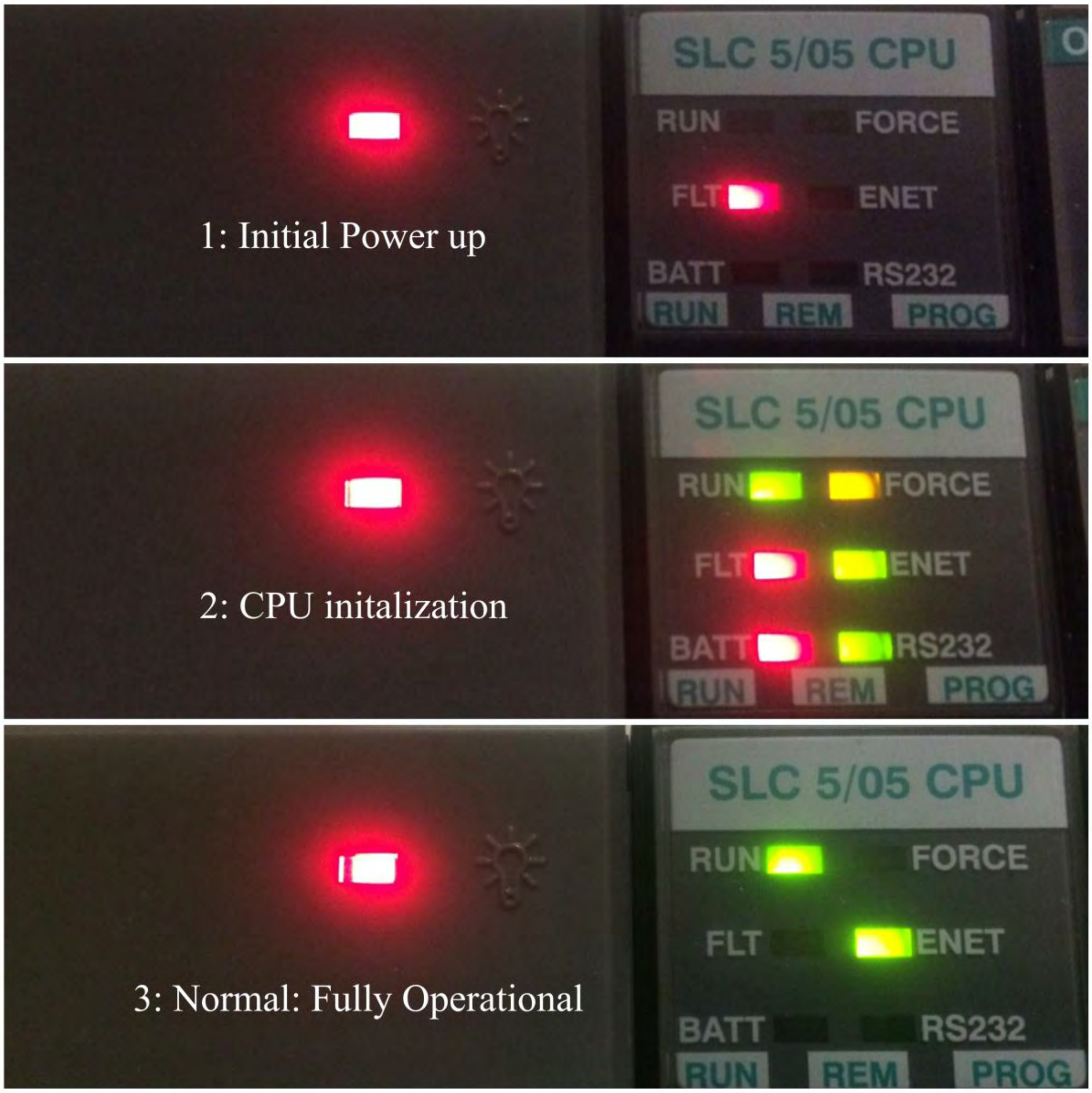


Figure F16. Sequence of indications on the PLC as it powers up

Also verify that the proximity sensors all have green LED lights lit on them indicating they have power and are NOT active (sensing an object). Once all the above indications and settings are complete and visible then the testbed is ready for the software setup chapter F3.

F3. SOFTWARE SETUP

Verify all hardware connections and settings are correct and chapter F1 and F2 are complete before proceeding.

Power on the engineering workstation labeled “Alice” and turn on the power strip shown in Figure F. Alice is a Windows 7 based machine that runs the window XP virtual machine containing the HMI and all software needed to run and maintain the testbed.

The following software is required to run the testbed:

- Windows 7 (Alice)
- VMware
- Windows XP (VM)
- RSView32 Runtime
- RSView32 Works
- RSlogix 500
- RSlinx (Professional)

Power up the Windows 7 machine (Alice) and start the VMware virtual machine titled “A2396” or “SCADA_AB_HMI.”

Once the windows XP virtual machine A2396 starts up, login using the user name and password provided. Next, verify that the PLC and VM are communicating by opening the RSLinx software found under the start menu under: Start -> Program files -> Rockwell Automation -> RSLinx.

Once RSlinx is open click on the “RSWho” (see Figure F17) icon and click the “+” button to expand the “AB_ETH-2, Ethernet” link and verify that there is not a red “X” over the PLC icon labeled “192.168.1.2, SLC-5/05, MINISSC”. If there is a red “X” over the PLC icon refer to the troubleshooting section (see Figures F17 and F18).

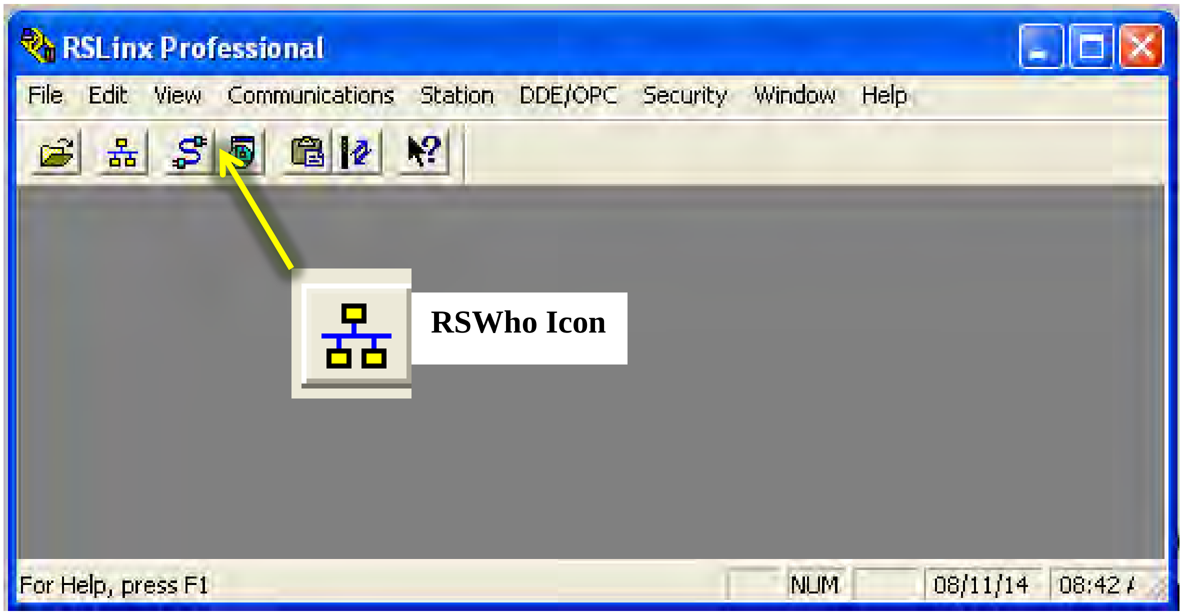
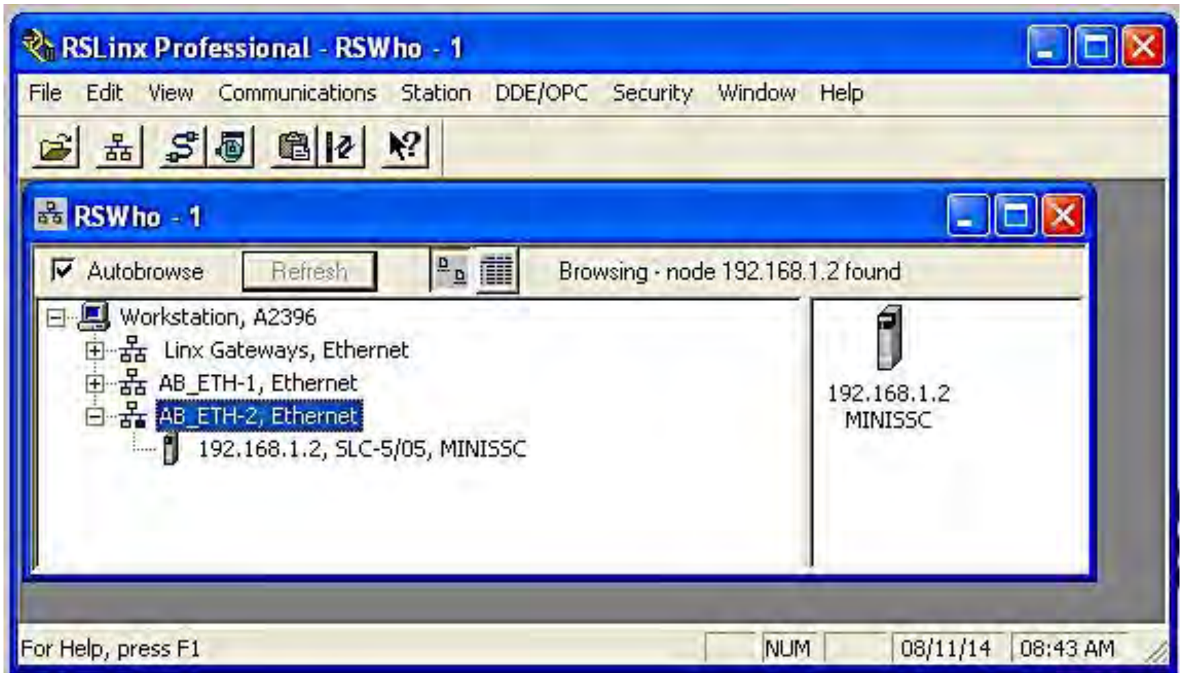

RSLinx Professional
File Edit View Communications Station DDE/OPC Security Window Help
RSWho Icon
For Help, press F1
NUM
08/11/14
08:42 A

Figure F17. RSLinkx initial screen with the RSWho icon


RSLinx Professional - RSWho - 1
File Edit View Communications Station DDE/OPC Security Window Help
RSWho - 1
☑ Autobrowse
Refresh
For Help, press F1
NUM
08/11/14
08:43 AM

Browsing - node 192.168.1.2 found

- Workstation, A2396
  - Linx Gateways, Ethernet
  - AB_ETH-1, Ethernet
  - AB_ETH-2, Ethernet
    - 192.168.1.2, SLC-5/05, MINISSC

192.168.1.2
MINISSC

Figure F18. RSLinx, RSWho communication status good

Verify the status of the PLC from the engineering workstation using the RSlogix 500 program by performing the following:

1. Open RSLogix 500 under Start -> Program files -> Rockwell Automation -> RSLogix 500 -> RSLogix 500.
2. Once the RSLogix 500 program is open click on File -> Open and choose the location of the lader logic located at My Documents -> AB_SCADA_TESTBED-> Ladder_Logic_AB_Testbed -> AB_SCADA_Testbed.RSS
3. Click the drop down menu upper left hand corner displaying "Offline" and choose "Go Online."
4. The drop down menu should turn Green and display "Remote Run," the ladder logic program appears and the ladder logic icon should start spinning.
    a. If the system is faulted the drop down menu will be RED and display the text "Faulted."
    b. If "Faulted" appears, then click the drop down menu arrow and choose "clear Fault" from the choices. The system should return to the green "Remote Run."
    c. If the system will not return to green "Remote Run" then re-verify your connections and configurations IAW sections F1 and F2 of this Appendix.
5. Once the green "Remote Run" is visible the PLC is ready and communicating with the engineering workstation and HMI.
6. Close RSlogix 500 by clicking File -> Exit.

Now that the PLC and HMI are communicating and there are no faults on the PLC, you are ready to open the HMI. To open the HMI double click on the RSView32 Runtime shortcut on the desktop titled "HMI_LAUNCH_AB_Testbed" (see Figure F19) located on the desktop or at the following file location:

C:\DocumentsandSettings\DEMO\MyDocuments\AB_SCADA_TESTBED\HMI_AB_SCADA_TESTBED\AB_TESTBED_HMI\AB_TESTBED_HMI.rsv

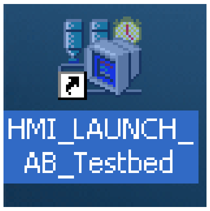


Figure F19. HMI shortcut icon

Once the HMI loads you will see a screen similar to Figure F20. From this screen you are ready to verify the HMI is in the "Ready State" for testing. The remainder of this chapter describes the settings required to achieve the Ready State.

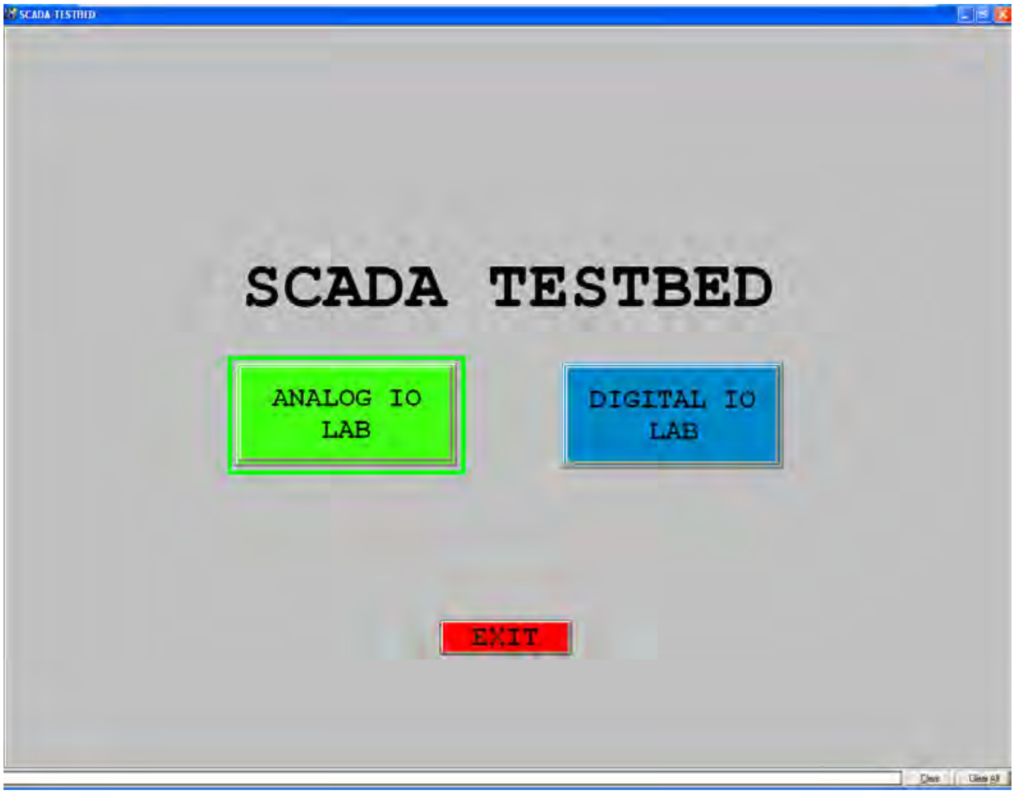
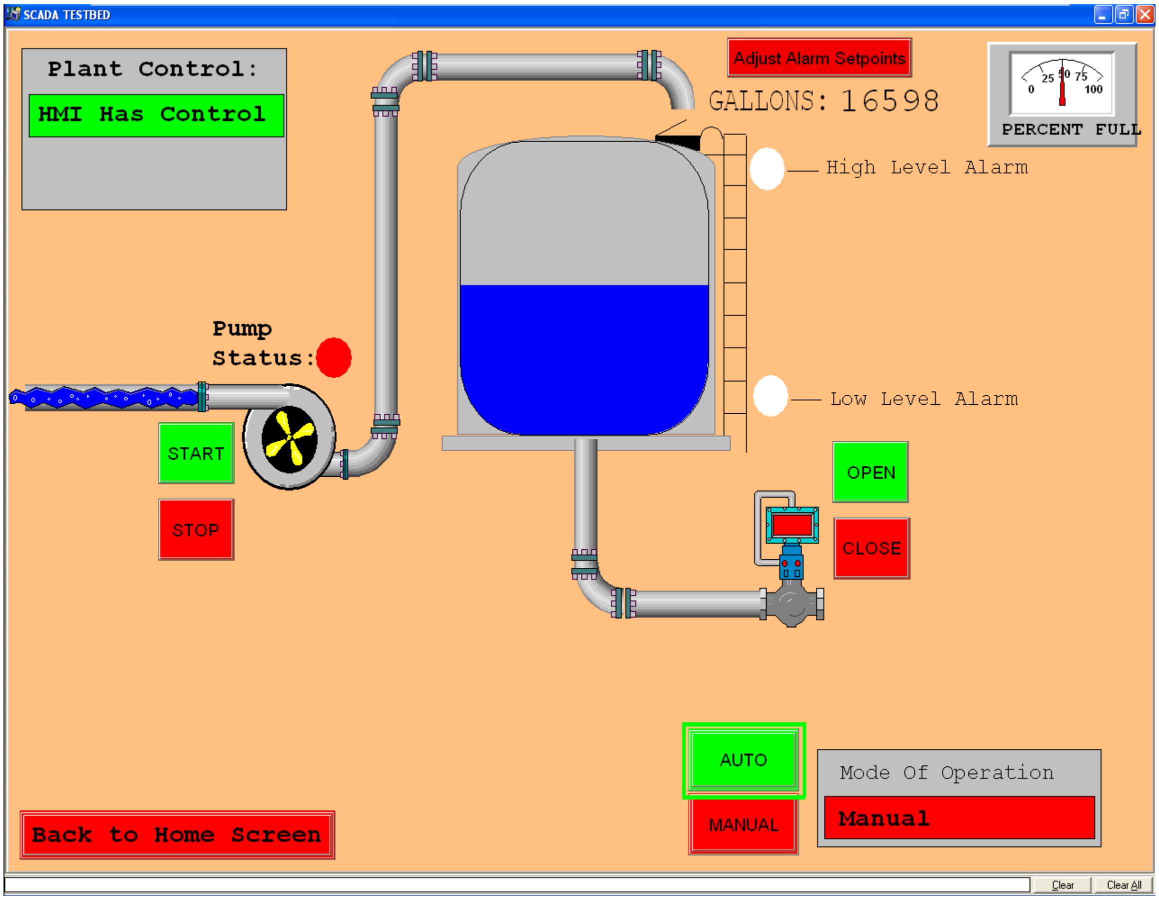


Figure F20. Testbed HMI home screen

Click on the green "ANALOG IO LAB" button and verify that the screen looks as follows:



Plant Control: HMI Has Control

Pump Status: ●

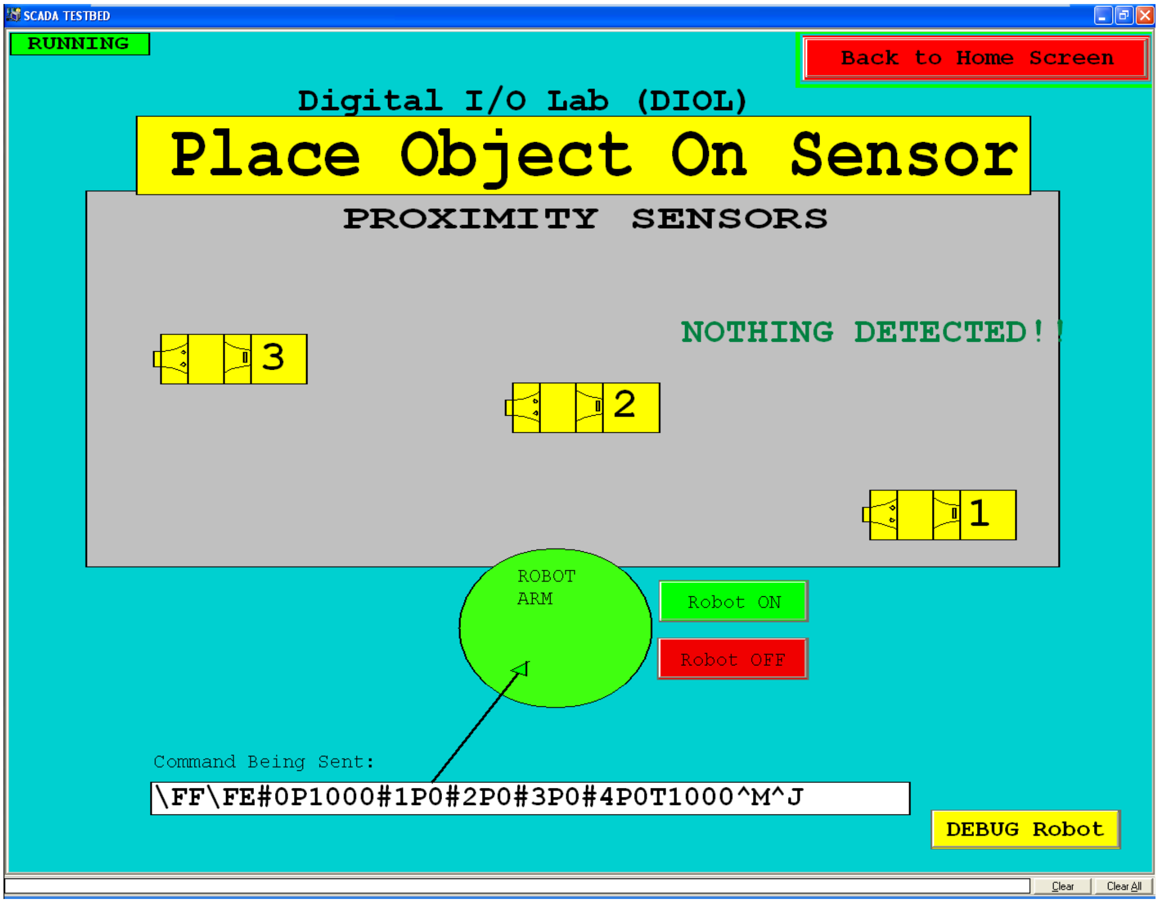
GALLONS: 16598

Mode Of Operation: Manual

Figure F21. Ready State of the AFTL as displayed at the HMI

Click on the red “Adjust Alarm Setpoints” button and set the high alarm level to “22000” gallons and the low alarm level setpoint to “5000” gallons. This will be the default alarm setpoints used in the Ready State. Click on the red “MANUAL” button to ensure the system is in Manual mode. Click on the green “Finished” button and close the alarm setpoint boxes.

Click on “Back to Home Screen” and click on the blue “DIGITAL IO LAB” button and verify the display looks as follows:



```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Figure F22. “Ready State” (and start state)of the DIOL as displayed at the HMI

After verification that the HMI screen looks like Figure F22, click on the “Back to Home Screen” button and return to the HMI “Home” screen as seen in Figure F20. The HMI and the entire lab are now in the “Ready State” and testing can begin. The Ready State will be state the lab will be in prior to, and after, each testing procedure in chapter F4.

F4. TESTING PROCEDURES

The following procedures will exercise all the functional tests outlined in chapter VI above and demonstrate that all the requirements set forth have been satisfied by the implemented testbed. Each test will begin and end in a “Ready State” as a baseline state for each test.

The “Ready State” is reached after completion of chapters F1-F3 of this Appendix. If doubt is raised as to improper setup or the testbed not being in the “Ready State” then all procedures or settings outlined in chapters F1-F3 should be revisited and then a re-test attempted. The order in which the test cases are conducted does not affect other tests, each test procedure result is independent of the others.

FUNCTIONAL TESTING

When reading the steps of each functional test the format for indicating which station an observation or action is performed is indicated by a preceding station name. For example: “HMI” “HOS” or “FLSD”. When no station is indicated during a step it is implied that directions are clear enough to discern how and where the action is to be conducted. For example “Place an object on the proximity sensors” does not have a station but is specific enough to understand the action.

Note: The fluid level may fluctuate as much as 700 gallons during testing without adjusting the FLSD. This is due to changing load to the PLC 24VDC power supply, is not an error, and should be expected.

TEST TF1: Test the proper display of the fluid level using the FLSD

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the green “ANALOG IO LAB” button.

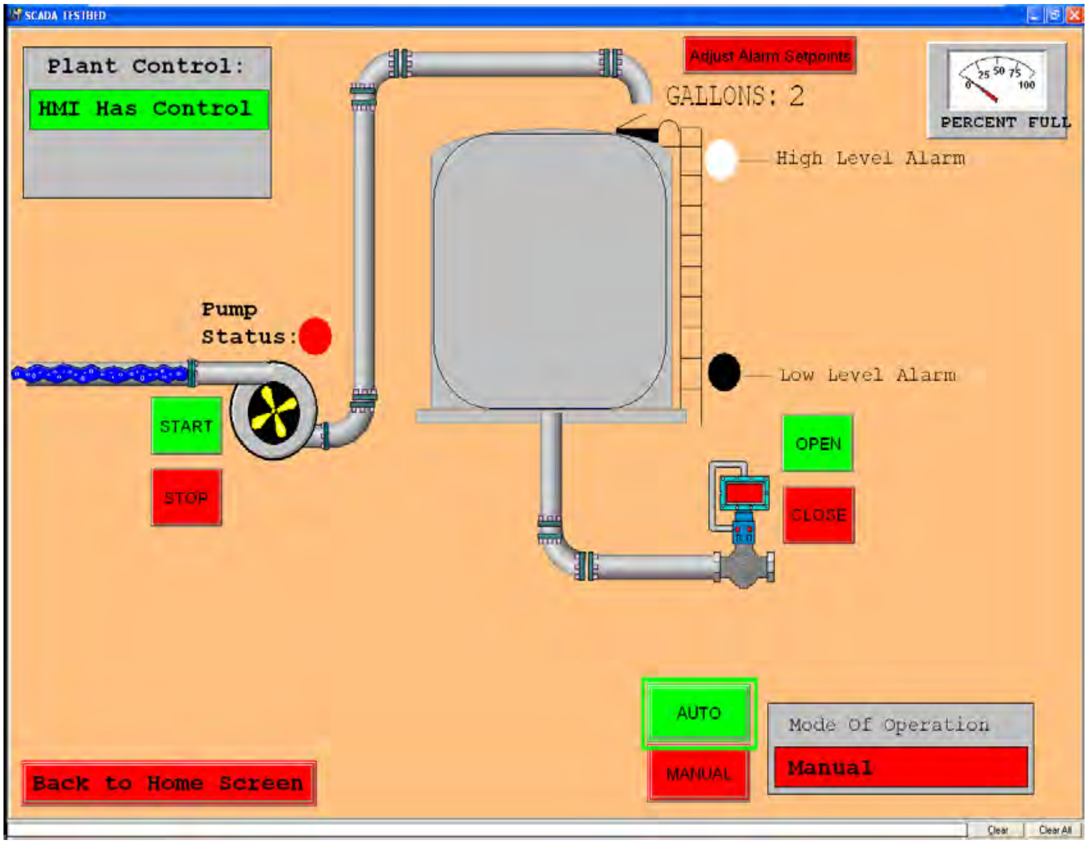
Step 3. HMI: click on the red “MANUAL” button and ensure the Mode of Operation is “Manual”

Step 4. FLSD: place the dial to the “0” position

Observe: The FLSD dial pointer is pointed at “0”

Observe: HMI: the level indicated on the fluid tank graphic is almost completely invisible and the gallons indicated are less than 5 gallons.

Observe: HMI: the low level alarm red indicator is flashing

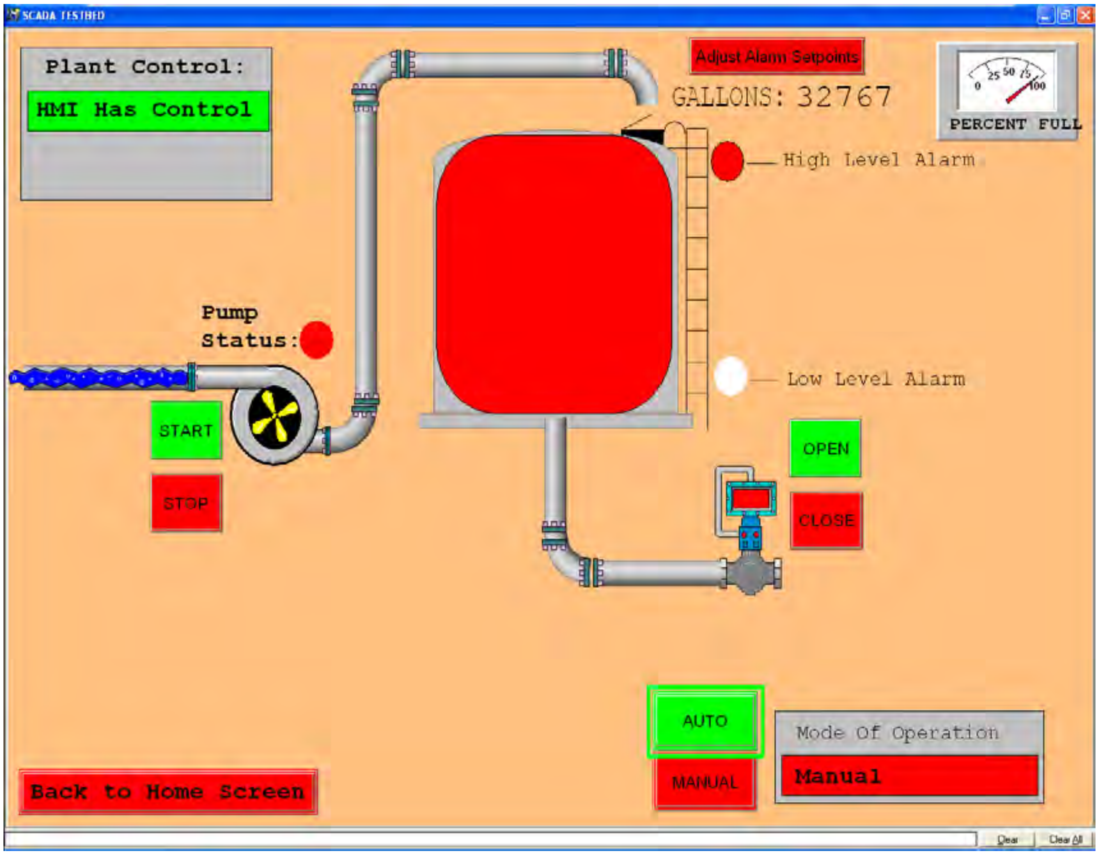


Step 5. FLSD: place the dial to the "100"

Observe: FLSD: dial pointer is pointed at "100"

Observe: HMI: the level indicated on the fluid tank graphic is full and the gallons indicated are greater than 32,500 gallons.

Observe: HMI: the High level alarm red indicator is flashing



Plant Control:
HMI Has Control

GALLONS: 32767

Mode Of Operation
Manual

Step 6. FLSD: continuously oscillate the dial between “0” and “100”.

Observe: HMI: using the help of another person or after turning a screen so the FLSD operator can observe, observe the fluid changing between “0” and “100” as the dial is moved.

Step 7. FLSD: stop oscillating the dial and place it at “50”

Step 8. HMI: click the “Back to Home Screen”

Observe: Lab has been placed back to the Ready State.

Observe: Test TF1 is complete

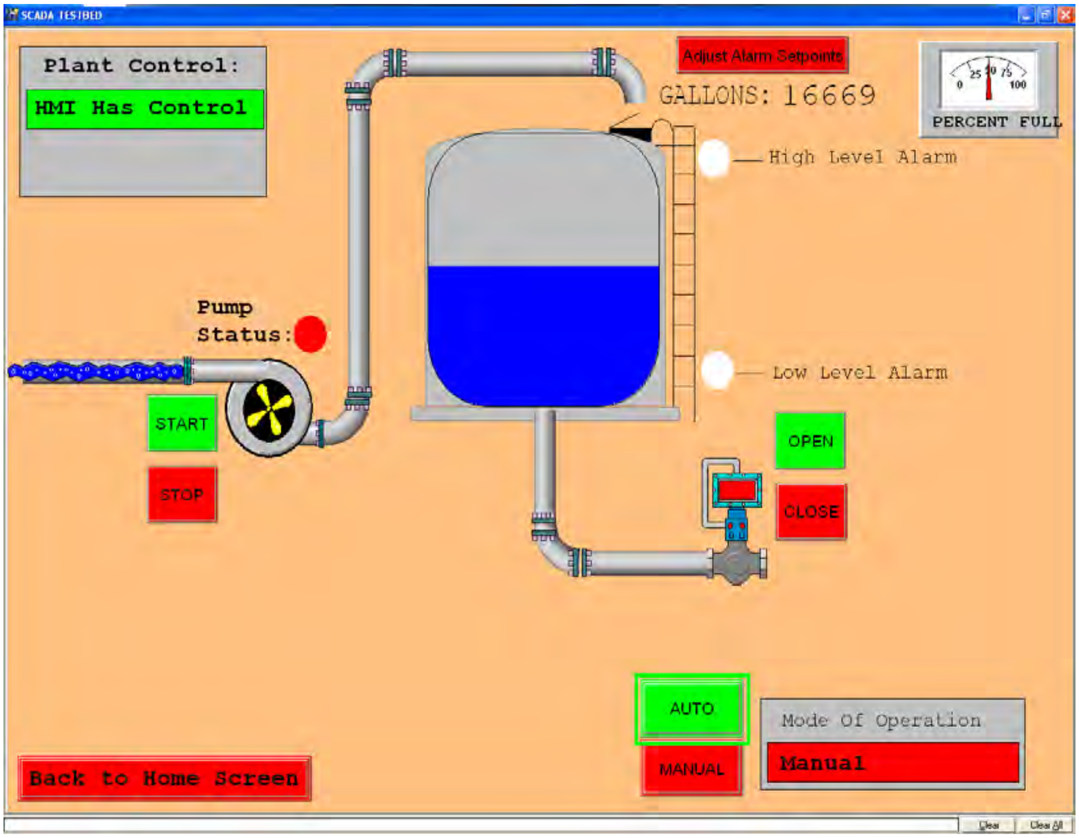
TEST TF2: Observe, start and stop the fill pump

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the green “ANALOG IO LAB” button.

Step 3. HMI: click on the red “MANUAL” button and ensure the Mode of Operation is “Manual”

Step 4. Click on the green “START” button next to the pump icon.



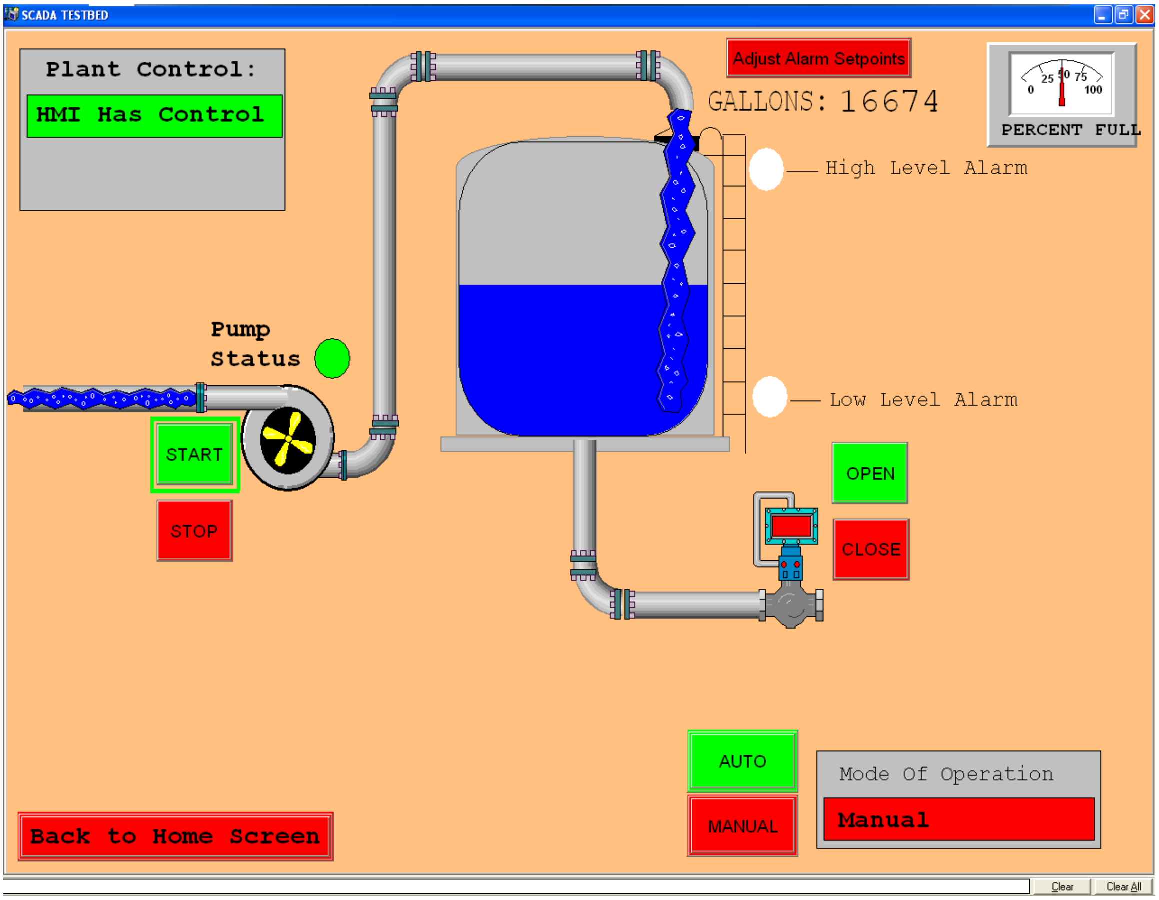
SCADA TESTBED

Plant Control: HMI Has Control

GALLONS: 16669

Mode Of Operation: Manual

Observe: The “Pump Status” indicator flashes green, the pump icon appears to be spinning and the blue water graphic appears to fill the tank.



Plant Control:
HMI Has Control

GALLONS: 16674

Mode Of Operation
Manual

Observe: HOS: the green LED labeled "PUMP ON" is illuminated.

Step 5. Click the red "STOP" pump button.

Observe: HMI: the "Pump Status" indicator returns to red and stops flashing, the pump icon stops spinning and the blue water fill graphic inside the tank disappears.

Observe: HOS: the green "PUMP ON" LED extinguishes

Step 6. HOS: place the "HMI Override switch" to "ON"

Step 7. HOS: place the "Fill Pump Control" switch in the "ON" position.

Observe: HOS: the green "PUMP ON" LED is lit.

Observe: HMI: the "Plant Control" graphic indicates "Override Station" in red indicating the HOS has control

Observe: HMI: the pump icon is spinning, the “Pump Status” indicator is flashing green and the blue water fill graphic is visible inside the tank.

Step 8. HOS: place the “Fill Pump Control” switch in the “OFF” position.

Observe: At the HOS: the green “PUMP ON” LED extinguishes.

Observe: At the HMI: the pump icon is no longer spinning, the “Pump Status” indicator is red and the blue water fill graphic is no longer visible inside the tank.

Step 9. HOS: place the “HMI Override Switch” to “OFF”

Observe: HMI: the “Plant Control” indicator indicates “HMI Has Control”

Step 10. HMI: click the red “Back to Home Screen” button

Observe: The lab has been returned to the Ready State

Observe: The test TF2 is complete

TEST TF3: Observe, open and close the drain valve

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the green “ANALOG IO LAB” button.

Step 3. HMI: click on the red “MANUAL” button and ensure the Mode of Operation is “Manual”

Step 4. HMI: Click on the green “OPEN” button next to the drain valve

Observe: HMI: the valve stem indictor flashes green and the blue water drain graphic appears

Observe: HOS: the green “VALVE OPEN” LED is lit.

Step 5. HMI: Click on the red “CLOSE” button next to the drain valve.

Observe: HMI: valve stem indictor turns red and the blue water drain graphic disappears.

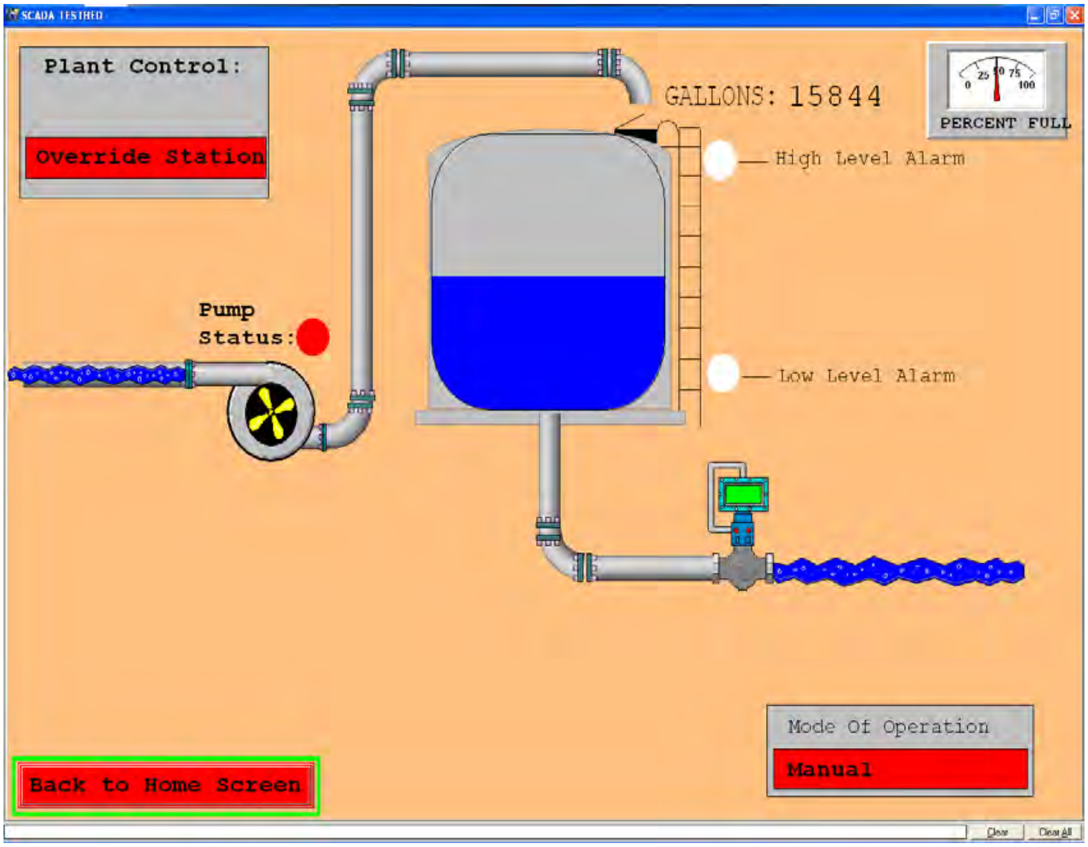
Observe: HOS: the green “VALVE OPEN” LED extinguishes.

Step 6. HOS: take control at the HOS by placing the “HMI Override Switch” to “ON”

Step 7. HOS: open the drain valve by placing the “DRAIN VLV CONTROL” switch to the “OPEN” position.

Observe: HOS: the green “VALVE OPEN” LED is lit.

Observe: HMI: the “Plant Control” indicator displays “Override Station”, the valve stem of the drain valve is flashing green and the blue water drain graphic is visible.



Step 8. HOS: close the drain valve by placing the "DRAIN VLV CONTROL" switch to the "CLOSED" position.

Observe: HOS: the green "VALVE OPEN" LED is extinguished.

Observe: HMI: the "Plant Control" indicator displays "Override Station", the valve stem of the drain valve is red and the blue water drain graphic is no longer visible.

Step 9. HOS: return control to the HMI by placing the "HMI Override Switch" to "OFF"

Observe: HMI: the "Plant Control" indicator indicates "HMI Has Control"

Step 10. HMI: click the red "Back to Home Screen" button

Observe: The lab has been returned to the Ready State

Observe: The test TF3 is complete

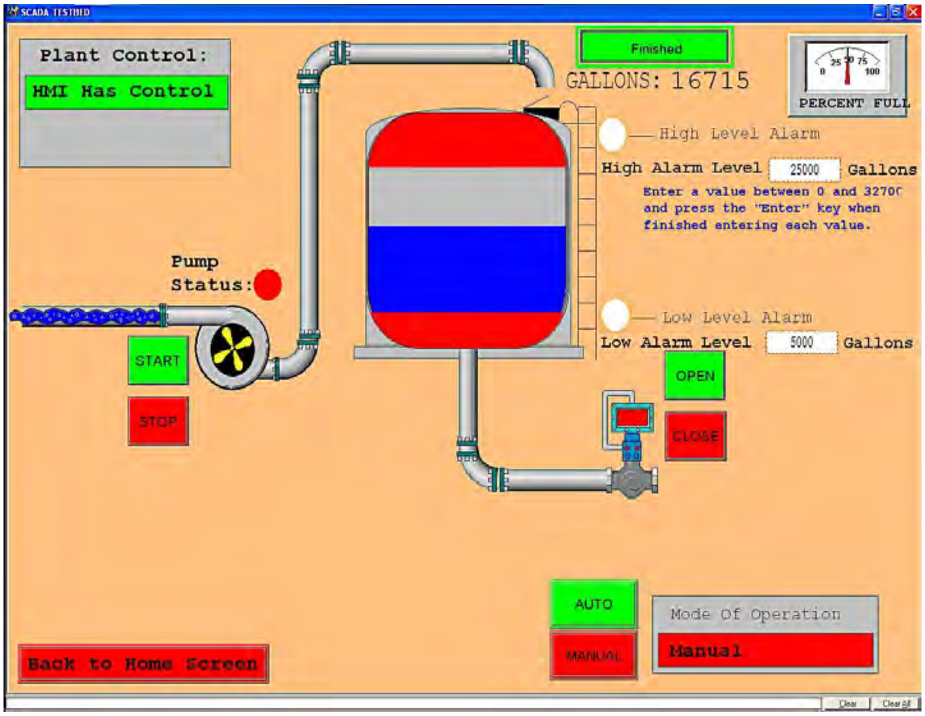
TEST TF4: Tank level alarm operation using the FLSD

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the green “ANALOG IO LAB” button.

Step 3. HMI: click on the red “MANUAL” button and ensure the Mode of Operation is “Manual”

Step 4. HMI: click on the red “Adjust Alarm Setpoints” button



Plant Control: HMI Has Control

GALLONS: 16715

High Alarm Level 25000 Gallons

Enter a value between 0 and 3270(and press the "Enter" key when finished entering each value.

Low Alarm Level 5000 Gallons

Mode Of Operation: Manual

Observe: HMI: the alarm level setpoints are 22000 and 5000

Step 5. HMI: Set the High Alarm Level to 25000 gallons and hit enter.

Step 6. HMI: click on the green “Finished” button

Observe: HMI: alarm setpoints disappear and the red level graphics on the fluid tank disappear.

Step 7. FLSD: adjust the dial to "70" %

Observe: HMI: No alarm is indicated and the displayed gallons is 23900 +/- 700.

Step 8. FLSD: adjust the dial to "80" %

Observe: HMI: a high fluid level alarm is indicated. The fluid level is flashing red and the "High Level Alarm" indicator is flashing red. Fluid level is above 25000 gallons.

Error Occured: If an alarm is not present, verify that the alarm setpoints are 25000 (high) and 5000 (low).

Step 9. FLSD: adjust the dial to "20"%

Observe: HMI: there is no alarm and the indicated fluid level is 6000 +/- 700 gallons.

Step 10. FLSD: adjust the dial to "15"%

Observe: HMI: a low fluid level alarm is indicated. The fluid level is flashing red and the "Low Level Alarm" indicator is flashing red. Fluid level is below 5000 gallons.

Error Occured: If an alarm is not present, verify the alarm setpoints are 25000 (high) and 5000 (low).

Step 11. FLSD: adjust the dial to "50"%

Observe: HMI: no alarms are present and the fluid level is approximately 17000 gallons.

Step 12. HMI: click the red "Back to Home Screen" button

Observe: The lab has been returned to the Ready State

Observe: The test TF4 is complete

TEST TF5: Tank level alarm set point adjustment

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the green “ANALOG IO LAB” button.

Step 3. HMI: click on the red “MANUAL” button and ensure the Mode of Operation is “Manual”

Step 4. HMI: click on the red “Adjust Alarm Setpoints” button

Step 5. HMI: click on the white High alarm level box and enter 19000 gallons and hit enter on the keyboard.

Observe: HMI: the red high level alarm setpiont graphic adjusts (enlarges) to display the new setpoint.

Step 6. HMI: click on the white Low Alarm Level box and change it to 10000 gallons and hit enter on the keyboard.

Observe: HMI: the red low level alarm setpoint graphic adjusts (enlarges) to display the new setoint.

Step 7. HMI: click the green “Finished” button

Step 8. FLSD: turn the dial to “30”%

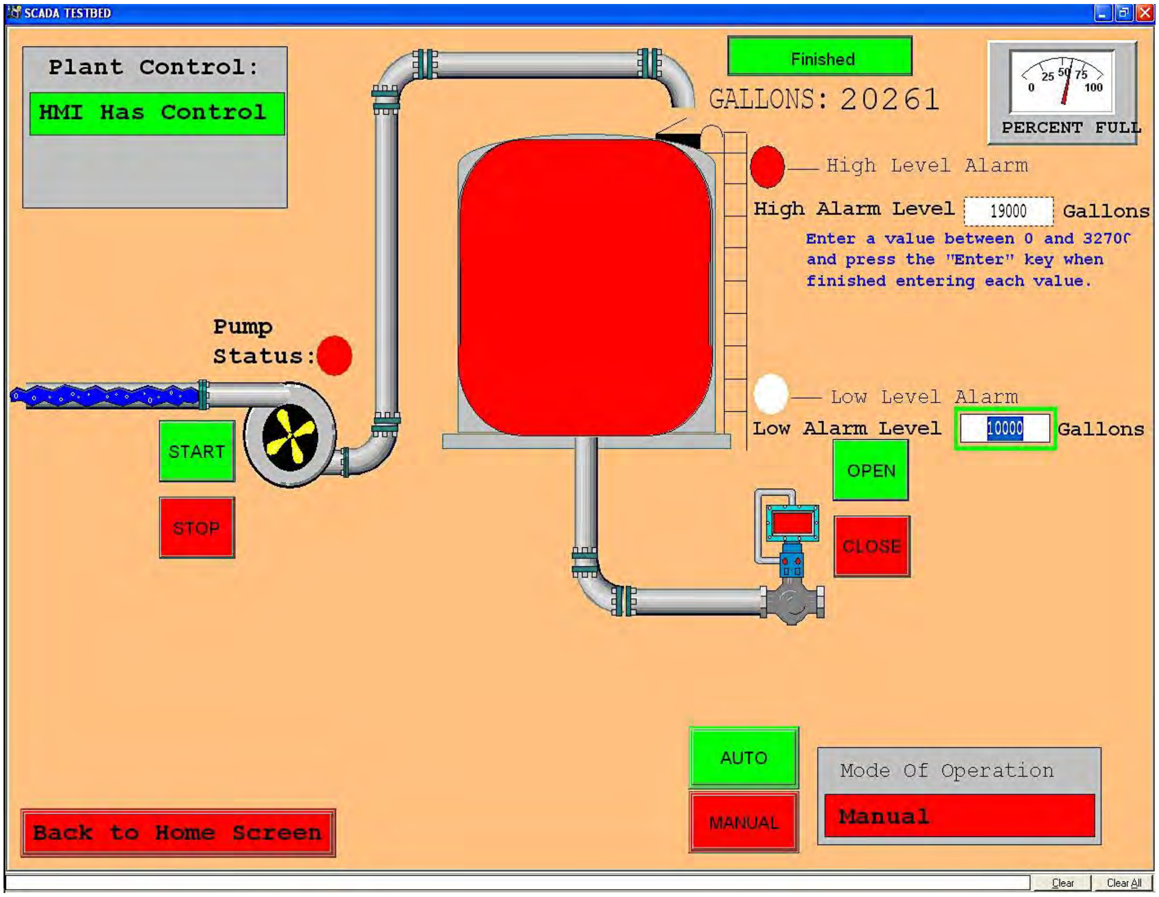
Observe: HMI: the Low Level Alarm indicator blinks red.

Observe: HOS: the red low level alarm LED is illuminated

Step 9. FLSD: turn the dial to “60”%

Observe: HMI: the High Level Alarm indicator blinks red and the fluid level blinks red.

Observe: HOS: the red high level alarm LED is illuminated



Plant Control: HMI Has Control

GALLONS: 20261

High Alarm Level 19000 Gallons

Enter a value between 0 and 3270
and press the "Enter" key when
finished entering each value.

Low Alarm Level 10000 Gallons

Mode Of Operation Manual

Step 10. FLSD: place the dial at “50”%

Observe: HMI: No alarms are visible

Step 11. HMI: Set the High Alarm Level to 22000 gallons and the Low Alarm Level to 5000 gallons.

Observe: HMI: red fluid alarm level graphics adjust to the new levels

Step 12. HMI: click the green “Finished” button

Observe: HMI: alarm setpoint adjustment boxes disappear

Step 13. HMI: click the red “Back to Home Screen” button

Observe: The lab has been returned to the Ready State

Observe: The test TF5 is complete

TEST TF6: Automatic mode operation

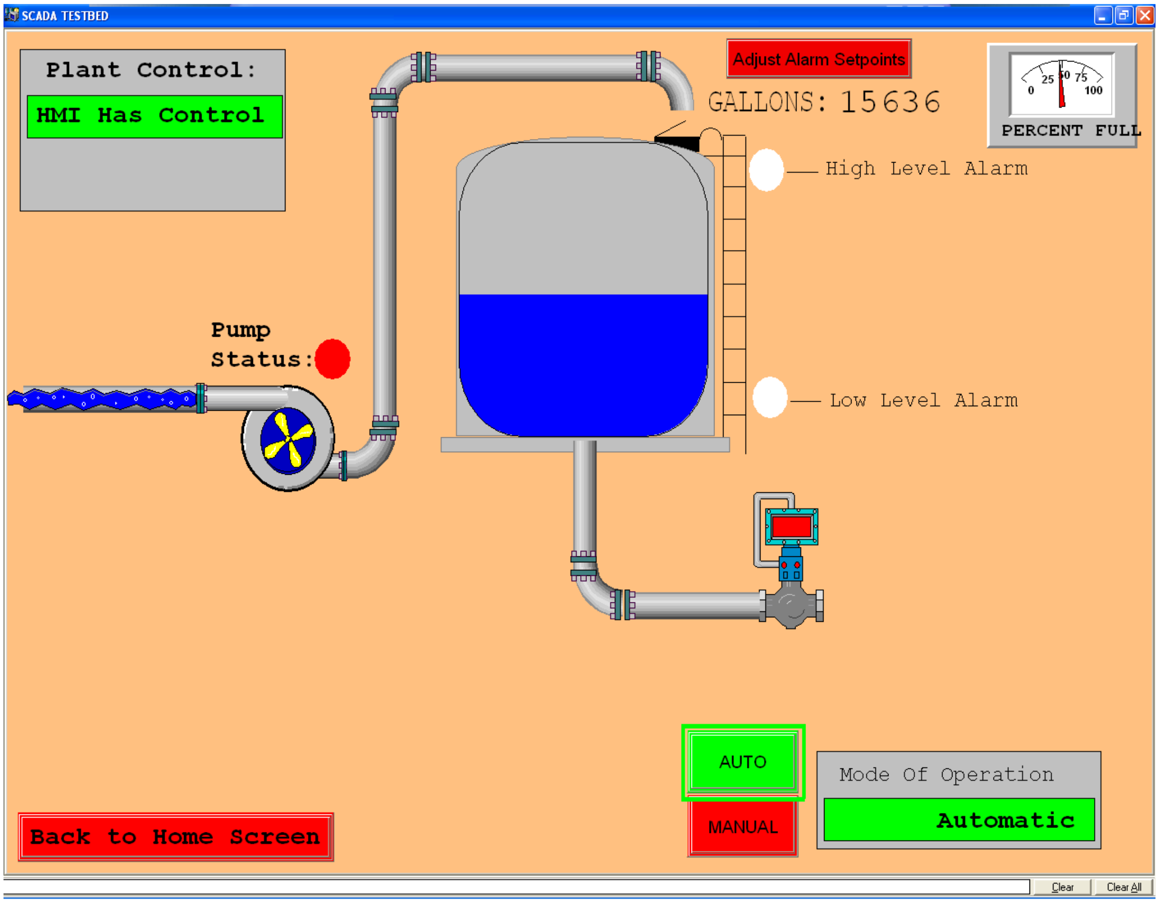
Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the green “ANALOG IO LAB” button.

Step 3. HMI: click on the green “AUTO” button

Observe: HMI: the displayed “Mode Of Operation” is green and displays “Automatic”

Observe: HMI: the button choices for the fill pump and the drain valve (on, off, open, close) disappear.



Plant Control: HMI Has Control

GALLONS: 15636

Mode Of Operation: Automatic

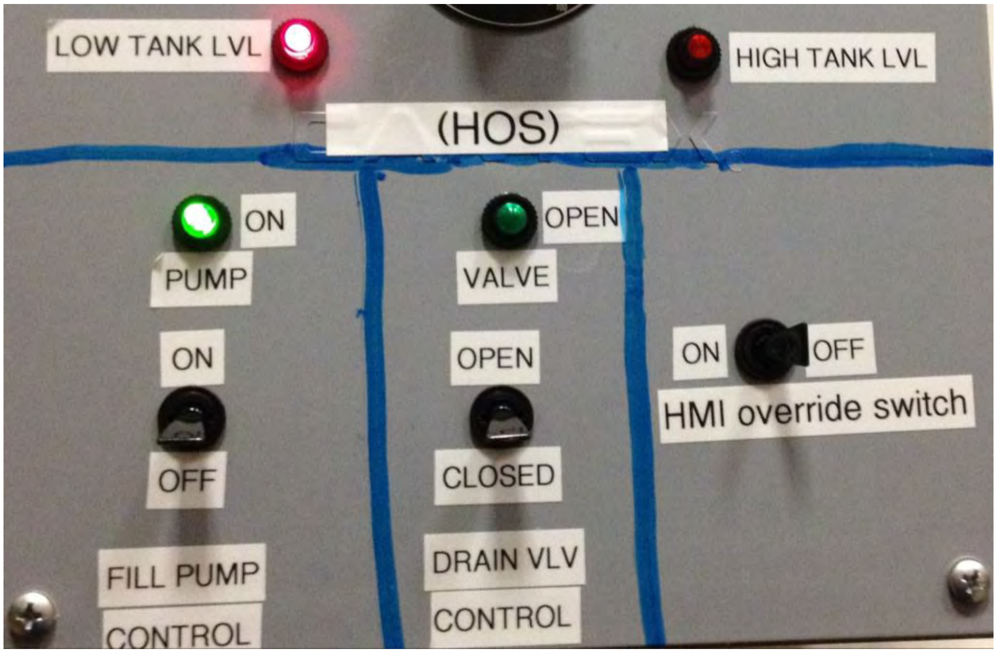
Step 4. FLSD: adjust the dial to “15”%

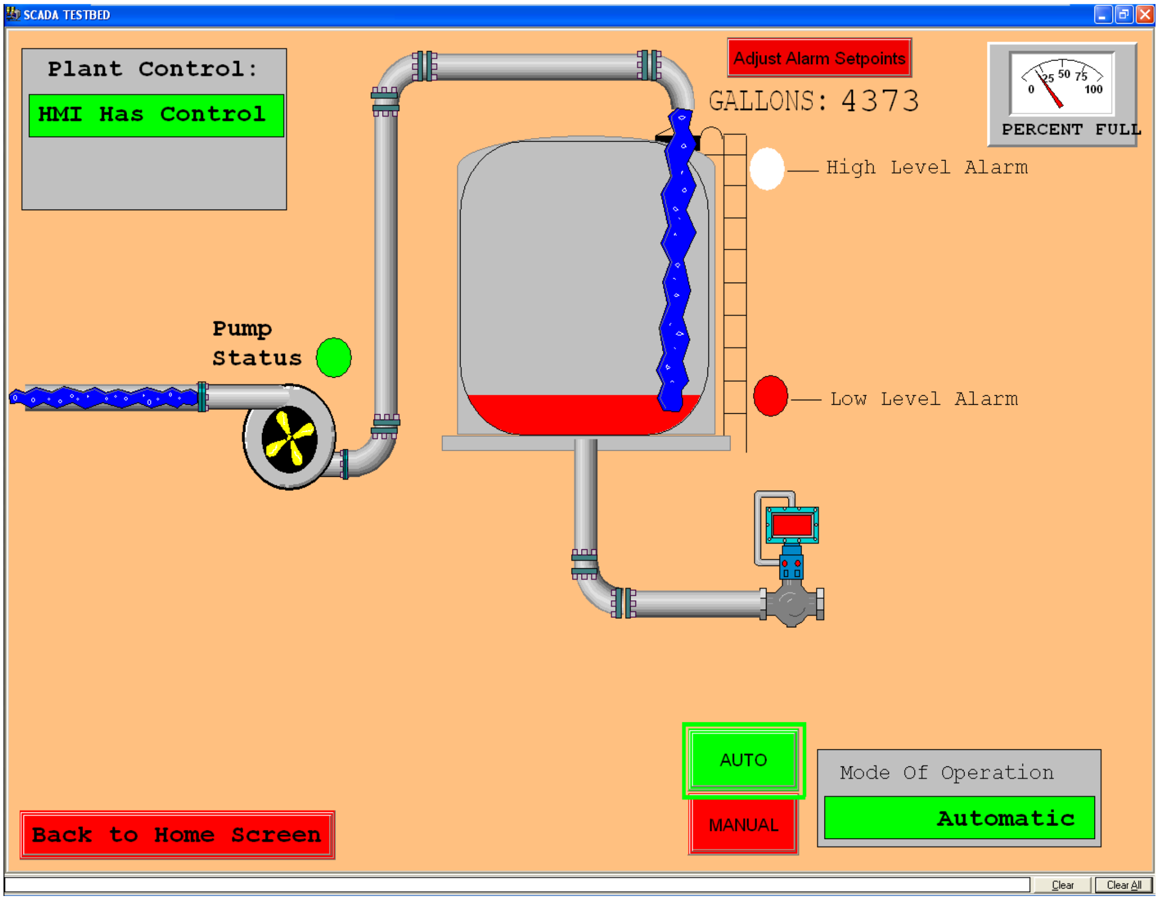
Observe: HMI: the Pump status icon is blinking green, the Pump icon is rotating, the Low Level Alarm graphic is blinking red, the fluid level is blinking red and the blue fill water graphic is visible inside the tank.

Observe: HMI: the drain valve stem is red, indicating the valve is still closed.

Observe: FLSD: the red "Low Tank LVL" LED is lit.

Observe: HOS: the green "PUMP ON" LED is lit





Plant Control: HMI Has Control

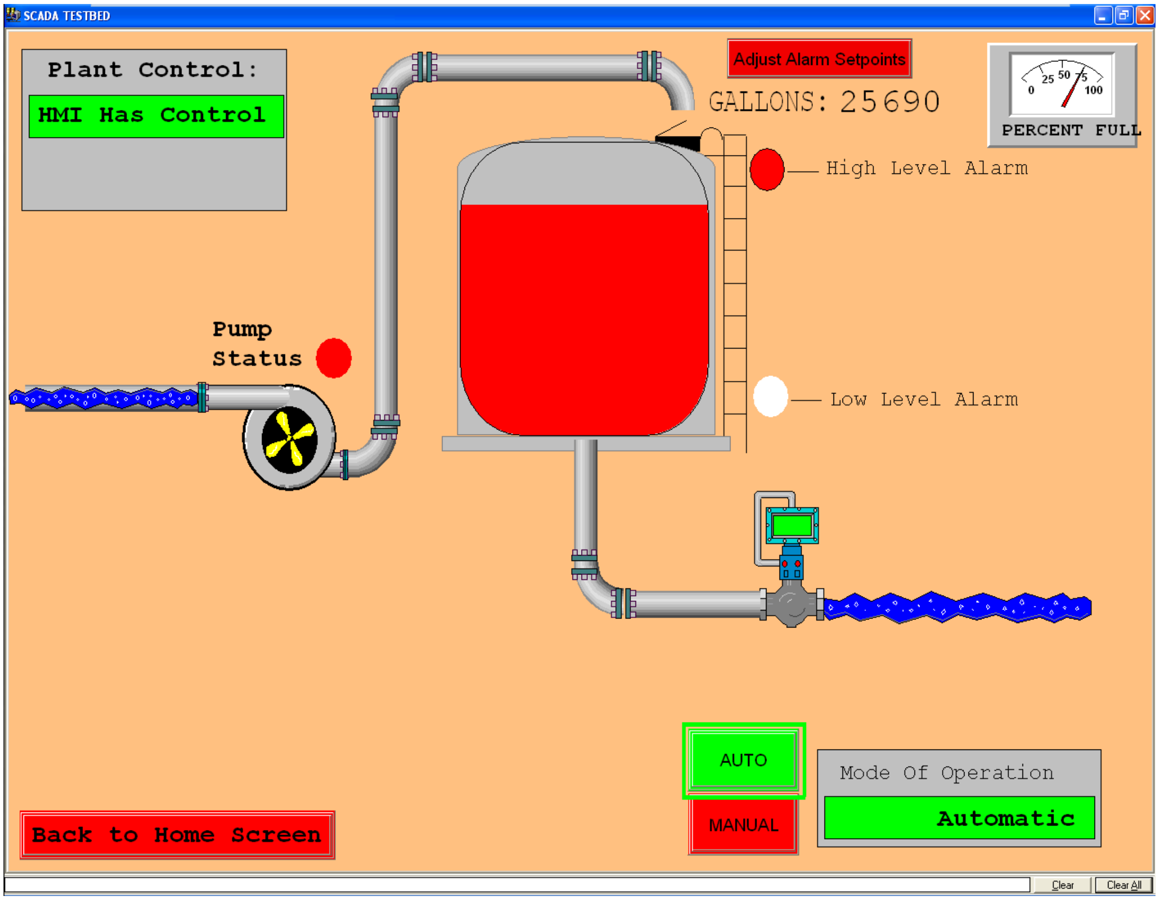
GALLONS: 4373

Mode Of Operation Automatic

Step 5. FLSD: adjust the dial to “75”%

Observe: HMI: the drain valve is open as indicated by a blinking green valve stem, the blue water drain graphic is visible, the High Level Alarm is blinking red and the fluid level inside the tank is blinking red.

Observe: HMI: the fill “Pump Status” indicator is red, indicating the pump is not running.



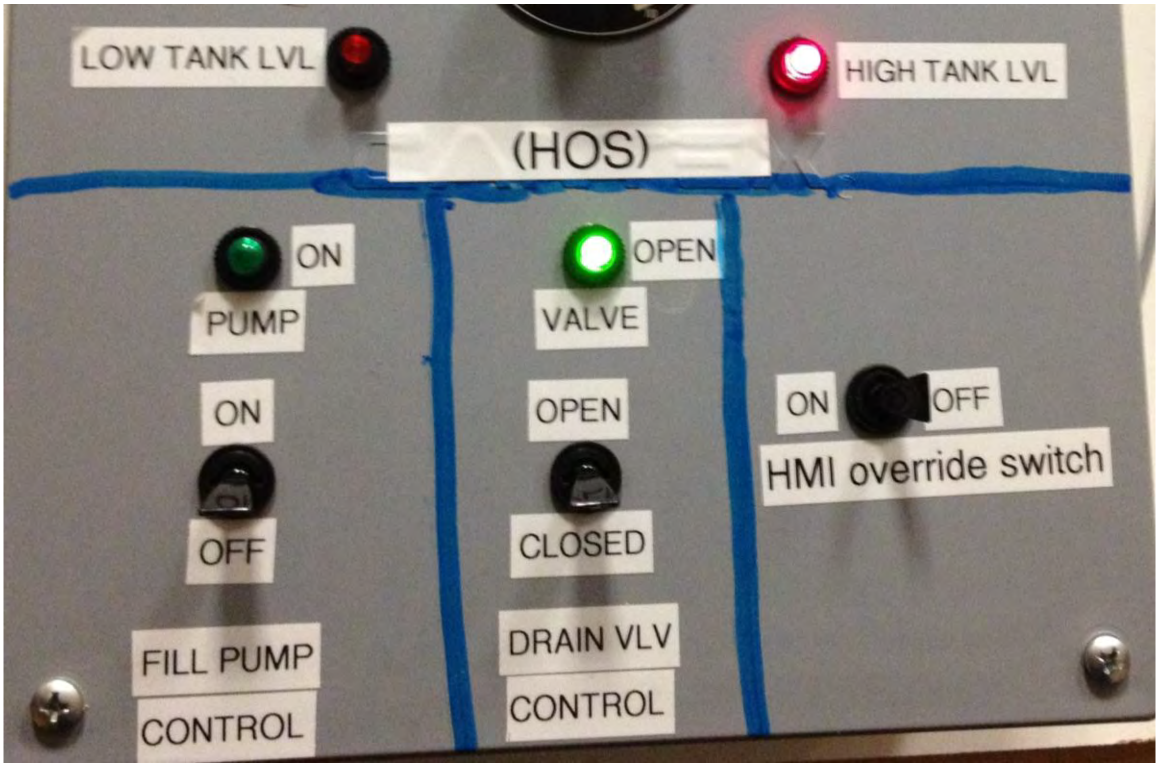
Plant Control: HMI Has Control

GALLONS: 25690

Mode Of Operation: Automatic

Observe: HOS: the green “VALVE OPEN” LED is lit and the green “PUMP ON” LED is extinguished.

Observe: FLSD: the red “High Tank LVL” LED is lit and the red “Low Tank LVL” LED is extinguished.

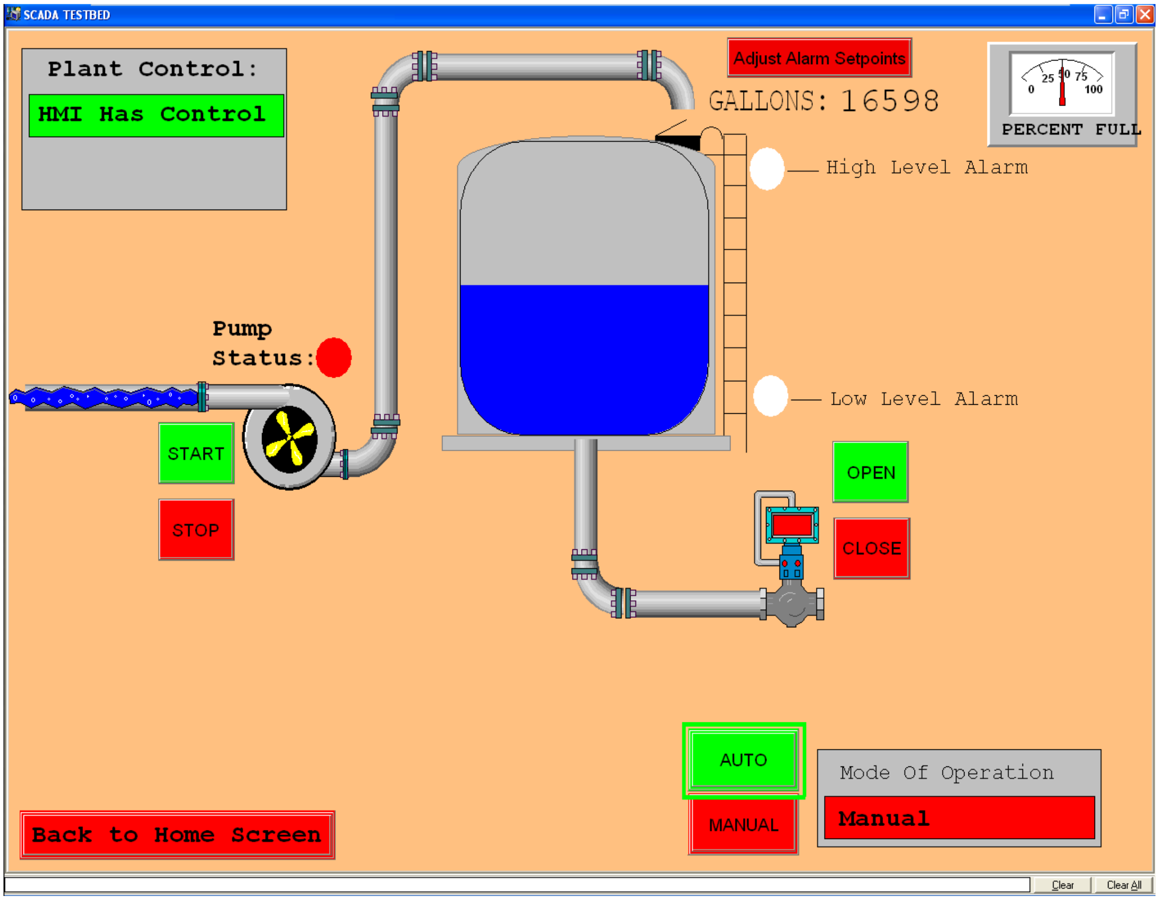


Step 6. FLSD: adjust the dial to “50”%

Observe: HMI: all alarms clear, the fill pump is off and the drain valve is closed.

Step 7. HMI: click on the red “MANUAL” button

Observe: HMI: “Mode Of Operation” indicates red “Manual” mode and the button choices for the pump and valve (on, off, open, close) appear.



Step 8. HMI: click the red "Back to Home Screen" button

Observe: The lab has been returned to the Ready State

Observe: The test TF6 is complete

TEST TF7: Override control functionality test

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the green "ANALOG IO LAB" button.

Step 3. HMI: click on the red "MANUAL" button and ensure the Mode of Operation is "Manual"

Step 4. HMI: click on the green "START" button for the fill pump

Observe: HMI: the fill pump starts, the pump graphic spins, the "Pump Status" graphic is blinking green and the blue fill water graphic is visible.

Observe: HOS: the green "PUMP ON" LED is lit

Step 5. HMI: click on the green "OPEN" button for the drain valve.

Observe: HMI: the valve is open, the valve stem color changes to blinking green and the blue water drain graphic is visible.

Observe: HOS: the green "VALVE OPEN" LED is lit

Step 6. HOS: place the "HMI Override Switch" to "ON"

Observe: HOS: the green "PUMP ON" and the "VALVE OPEN" LED's extinguish (indicating the HOS switch has control).

Observe: HMI: the fill pump and drain valve indicate "OFF" and "CLOSED" (respectively); the "Pump Status" graphic is red, the pump graphic is no longer spinning, the blue fill water and drain graphics are no longer visible and the drain valve stem is red.

Observe: HMI: the "Plant Control: indicator now displays the red "Override Station" indicating the HOS has control.

Step 7. HOS: open the drain valve by placing the "DRAIN VLV CONTROL" switch to the "OPEN" position.

Observe: HOS: the green "VALVE OPEN" LED is lit.

Observe: HMI: the drain valve is open; the valve stem of the drain valve is flashing green and the blue water drain graphic is visible. Additionally, the "Plant Control" indicator still displays "Override Station"

Step 8. HOS: close the drain valve by placing the "DRAIN VLV CONTROL" switch to the "CLOSED" position.

Observe: HOS: the green "VALVE OPEN" LED is extinguished.

Observe: HMI: the drain valve is closed; the valve stem of the drain valve is red and the blue water drain graphic is no longer visible. Additionally, the "Plant Control" indicator still displays "Override Station"

Step 9. HOS: place the "Fill Pump Control" switch in the "ON" position.

Observe: HOS: the green "PUMP ON" LED is lit.

Observe: HMI: The pump is on; the pump icon is spinning, the "Pump Status" indicator is flashing green and the blue water fill graphic is visible inside the tank. Additionally, the "Plant Control" indicator still displays "Override Station"

Step 10. HOS: place the "Fill Pump Control" switch in the "OFF" position.

Observe: At the HOS: the green "PUMP ON" LED extinguishes.

Observe: At the HMI: the pump is off; the pump icon is no longer spinning, the "Pump Status" indicator is red and the blue water fill graphic is no longer visible inside the tank. Additionally, the "Plant Control" indicator still displays "Override Station"

Step 11. HOS: return control to the HMI by placing the "HMI Override Switch" to "OFF"

Observe: HMI: the fill pump is running, the valve is open, HMI has control; the fill pump icon is spinning, the "Pump Status" icon is blinking green, the blue water fill graphic is visible, the drain valve stem is blinking green, the blue water drain graphic is visible and the “Plant Control” indicator indicates “HMI Has Control”

Observe: HOS: Valve Open and Pump ON LED’s are lit.

Step 12. HMI: Stop the fill pump and close the drain valve; click on the red “STOP” button to stop the fill pump, click on the red “CLOSE” button to close the drain valve.

Observe: HMI: the fill pump stops and the drain valve closes; the fill pump icon stops spinning, the “Pump Status” icon is red, the blue water fill graphic is no longer visible, the drain valve stem is red, the blue water drain graphic is no longer visible and the “Plant Control” indicator still indicates “HMI Has Control”

Observe: HOS: Valve Open and Pump ON LED’s are extinguished.

Step 13. HMI: click the red “Back to Home Screen” button

Observe: The lab has been returned to the Ready State

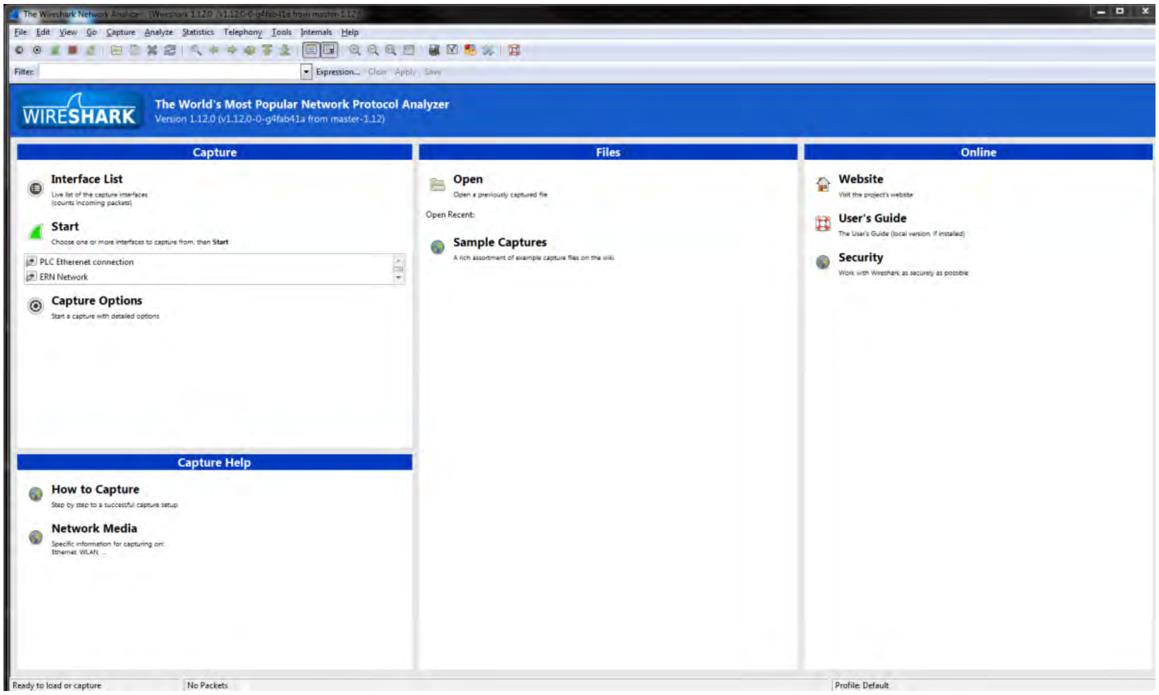
Observe: The test TF7 is complete

TEST TF8: Ethernet traffic monitoring

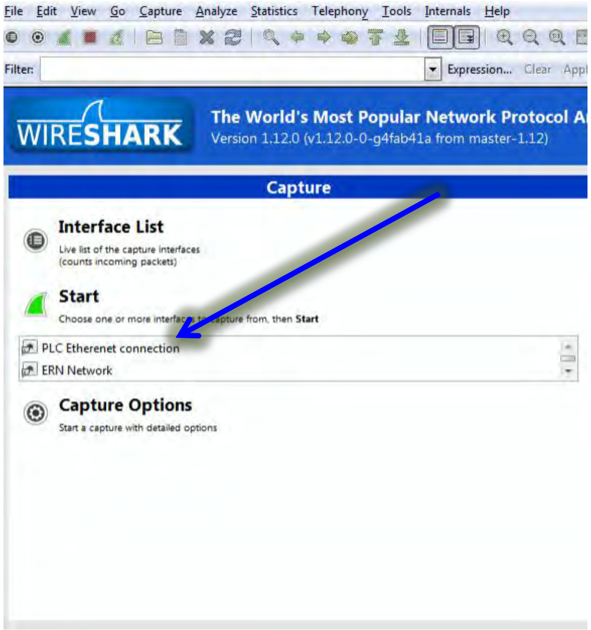
Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. (Optional) Engineering Workstation: open “Wireshark”; click on Start-> All programs -> Wireshark

Observe: The Wireshark program starts up and the main start screen is displayed.


The Wireshark Network Analyzer [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)
Capture
Interface List
Live list of the capture interfaces (counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
PLC Etherenet connection
ERN Network
Capture Options
Start a capture with detailed options
Capture Help
How to Capture
Step by step to a successful capture setup
Network Media
Specific information for capturing on: Ethernet, WLAN, ...
Files
Open
Open a previously captured file
Open Recent:
Sample Captures
A rich assortment of example capture files on the wiki
Online
Website
Visit the project's website
User's Guide
The User's Guide (local version, if installed)
Security
Work with Wireshark as securely as possible
Ready to load or capture
No Packets
Profile: Default

Step 3. Select the “PLC Ethernet connection” interface from the Capture section


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Appl
WIRESHARK
The World's Most Popular Network Protocol A
Version 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)
Capture

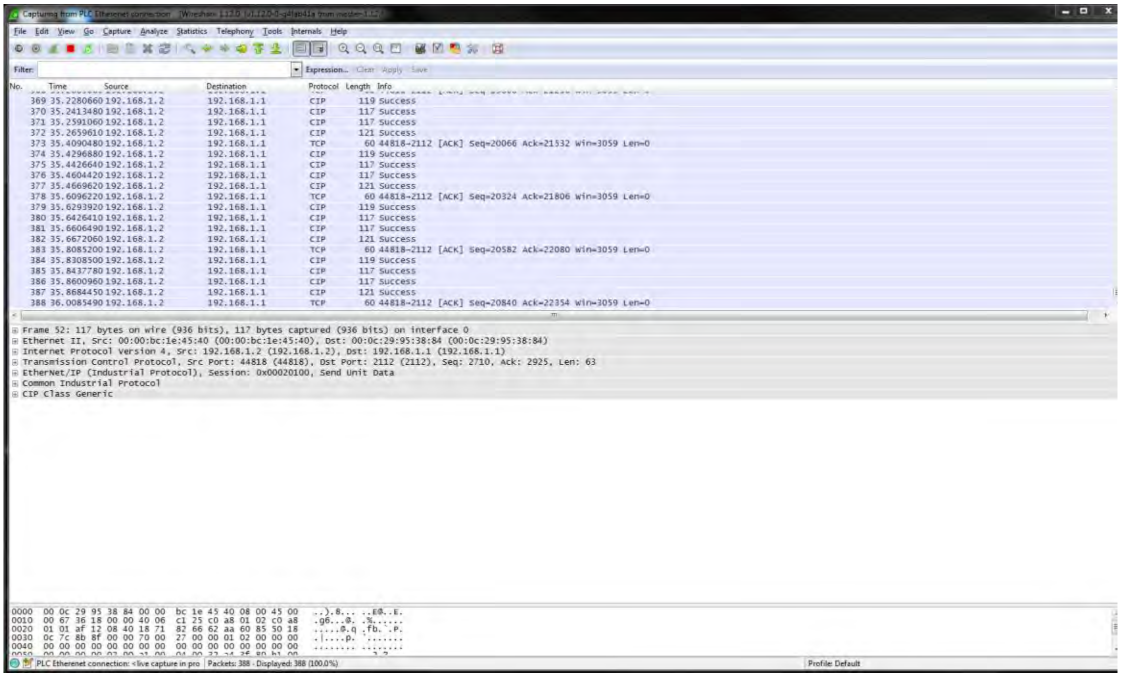
Interface List
Live list of the capture interfaces (counts incoming packets)

Start
Choose one or more interfaces to capture from, then **Start**

PLC Etherenet connection
ERN Network

Capture Options
Start a capture with detailed options

Step 4. Click the green Start (Shark fin) icon and being Ethernet packet capture of the PLC – HMI interface.


Capturing from PLC Etherenet connection [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.  Time       Source       Destination  Protocol Length Info
369 35.2280660 192.168.1.2   192.168.1.1   CIP    119 Success
370 35.2413480 192.168.1.2   192.168.1.1   CIP    117 Success
371 35.2591060 192.168.1.2   192.168.1.1   CIP    117 Success
372 35.2659610 192.168.1.2   192.168.1.1   CIP    121 Success
373 35.4090480 192.168.1.2   192.168.1.1   TCP     60 44818→2112 [ACK] Seq=20066 Ack=21532 Win=3059 Len=0
374 35.4296880 192.168.1.2   192.168.1.1   CIP    119 Success
375 35.4426640 192.168.1.2   192.168.1.1   CIP    117 Success
376 35.4604420 192.168.1.2   192.168.1.1   CIP    117 Success
377 35.4669620 192.168.1.2   192.168.1.1   CIP    121 Success
378 35.6096220 192.168.1.2   192.168.1.1   TCP     60 44818→2112 [ACK] Seq=20324 Ack=21806 Win=3059 Len=0
379 35.6293920 192.168.1.2   192.168.1.1   CIP    119 Success
380 35.6426410 192.168.1.2   192.168.1.1   CIP    117 Success
381 35.6606490 192.168.1.2   192.168.1.1   CIP    117 Success
382 35.6672060 192.168.1.2   192.168.1.1   CIP    121 Success
383 35.8085200 192.168.1.2   192.168.1.1   TCP     60 44818→2112 [ACK] Seq=20582 Ack=22080 Win=3059 Len=0
384 35.8308500 192.168.1.2   192.168.1.1   CIP    119 Success
385 35.8437780 192.168.1.2   192.168.1.1   CIP    117 Success
386 35.8600960 192.168.1.2   192.168.1.1   CIP    117 Success
387 35.8684450 192.168.1.2   192.168.1.1   CIP    121 Success
388 36.0085490 192.168.1.2   192.168.1.1   TCP     60 44818→2112 [ACK] Seq=20840 Ack=22354 Win=3059 Len=0
```

```
⊞ Frame 52: 117 bytes on wire (936 bits), 117 bytes captured (936 bits) on interface 0
⊞ Ethernet II, Src: 00:00:bc:1e:45:40 (00:00:bc:1e:45:40), Dst: 00:0c:29:95:38:84 (00:0c:29:95:38:84)
⊞ Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.1.1 (192.168.1.1)
⊞ Transmission Control Protocol, Src Port: 44818 (44818), Dst Port: 2112 (2112), Seq: 2710, Ack: 2925, Len: 63
⊞ EtherNet/IP (Industrial Protocol), Session: 0x00020100, Send Unit Data
⊞ Common Industrial Protocol
⊞ CIP Class Generic
```

```
0000  00 0c 29 95 38 84 00 00  bc 1e 45 40 08 00 45 00   ..).8... ..E@..E.
0010  00 67 36 18 00 00 40 06  c1 25 c0 a8 01 02 c0 a8   .g6...@. .%......
0020  01 01 af 12 08 40 18 71  82 66 62 aa 60 85 50 18   .....@.q .fb.`.P.
0030  0c 7c 8b 8f 00 00 70 00  27 00 00 01 02 00 00 00   .|....p. '.......
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

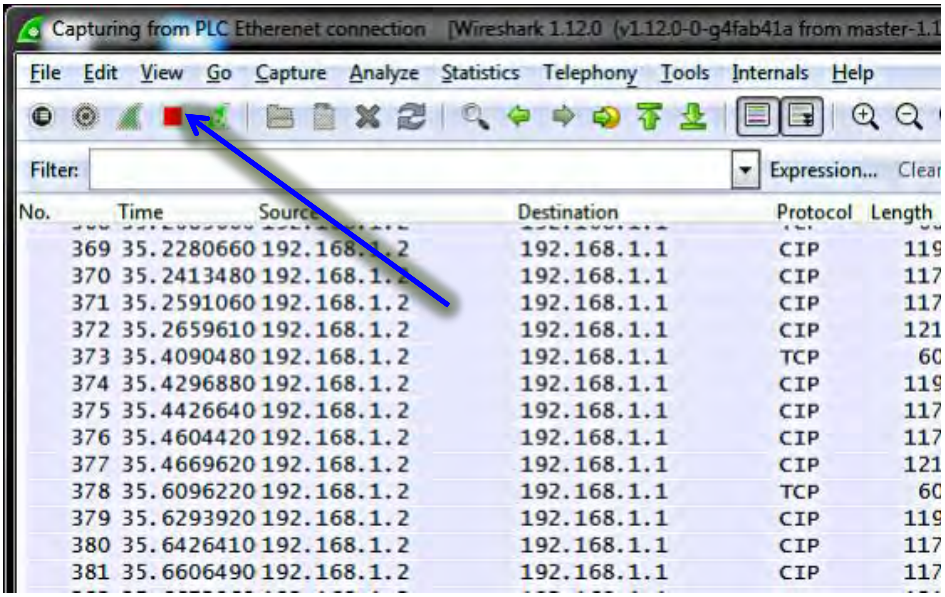
PLC Etherenet connection: <live capture in pro
Packets: 388 · Displayed: 388 (100.0%)
Profile: Default

Observe: The packet list screen will appear and allow for deeper packet inspection and analysis as desired.

Observe: New packets are being captured between IP addresses 192.168.1.1 and 192.168.1.2. (PLC and HMI)

Step 5. Engineering Workstation: Conduct all desired capture operations using the Wireshark interface.

Step 6. Engineering Workstation: Stop capturing and close the Wireshark program; click on the red square icon in the upper left of the capture screen.


Capturing from PLC Etherenet connection [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.1
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear

No.	Time	Source	Destination	Protocol	Length
369	35.2280660	192.168.1.2	192.168.1.1	CIP	119
370	35.2413480	192.168.1.2	192.168.1.1	CIP	117
371	35.2591060	192.168.1.2	192.168.1.1	CIP	117
372	35.2659610	192.168.1.2	192.168.1.1	CIP	121
373	35.4090480	192.168.1.2	192.168.1.1	TCP	60
374	35.4296880	192.168.1.2	192.168.1.1	CIP	119
375	35.4426640	192.168.1.2	192.168.1.1	CIP	117
376	35.4604420	192.168.1.2	192.168.1.1	CIP	117
377	35.4669620	192.168.1.2	192.168.1.1	CIP	121
378	35.6096220	192.168.1.2	192.168.1.1	TCP	60
379	35.6293920	192.168.1.2	192.168.1.1	CIP	119
380	35.6426410	192.168.1.2	192.168.1.1	CIP	117
381	35.6606490	192.168.1.2	192.168.1.1	CIP	117

Observe: The Wireshark program stops capturing packets.

Step 7. Engineering Workstation: Select from the menu bar File-> Quit and close Wireshark

Step 8. Decide if you want to save the capture, if so choose a location.

Observe: The Wireshark program will close after you have saved the packet capture or decided to exit without saving.

Step 9. Engineering Workstation: select the VMware program and return to the HMI and the testlab home screen.

Observe: Test TF8 complete

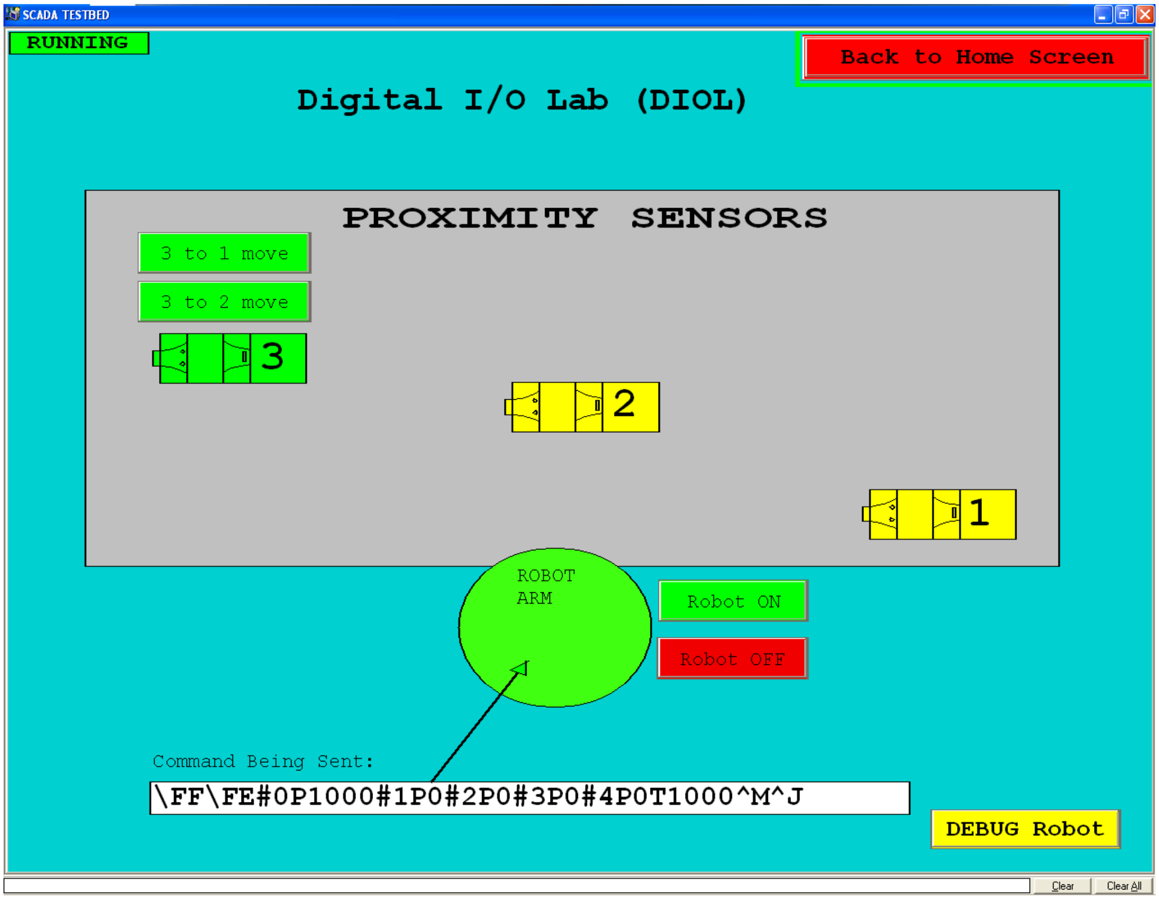
TEST TF9: Move objects between proximity sensors using the robot arm

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the blue “DIGITAL IO LAB” button.

Step 3. Place an object on proximity sensor #3

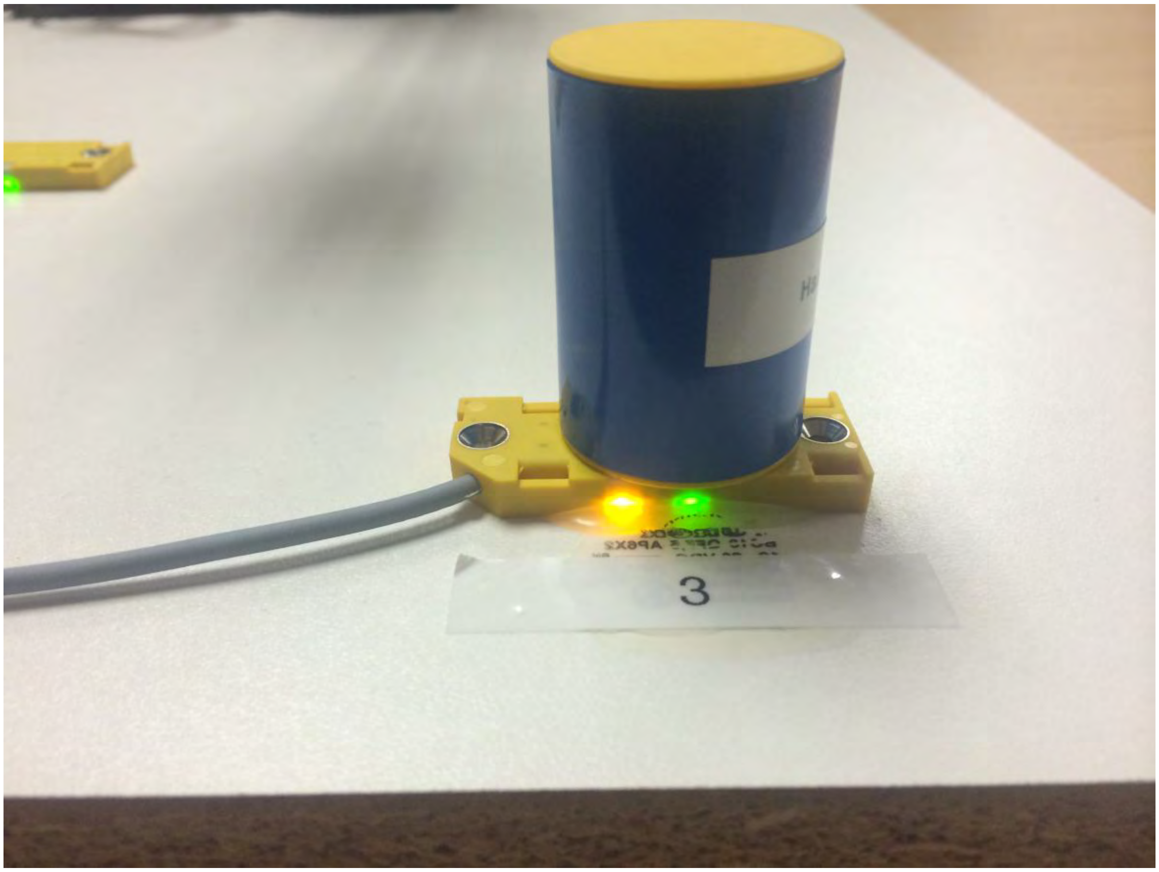
Observe: HMI: green move buttons appear as choices for the robot arm.



Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observe: Proximity Sensor: both the green and yellow LED indicators located on the sensor #3 are illuminated



Step 4. HMI: click on the green “Robot ON” button

Observe: Robot arm moves to the “stow” position. The stow position is when all the servos on the arm are moved to the 1500 position by the command “#0P1500#1P1500#2P1500#3P1500#4P1500T1000^M^J”.

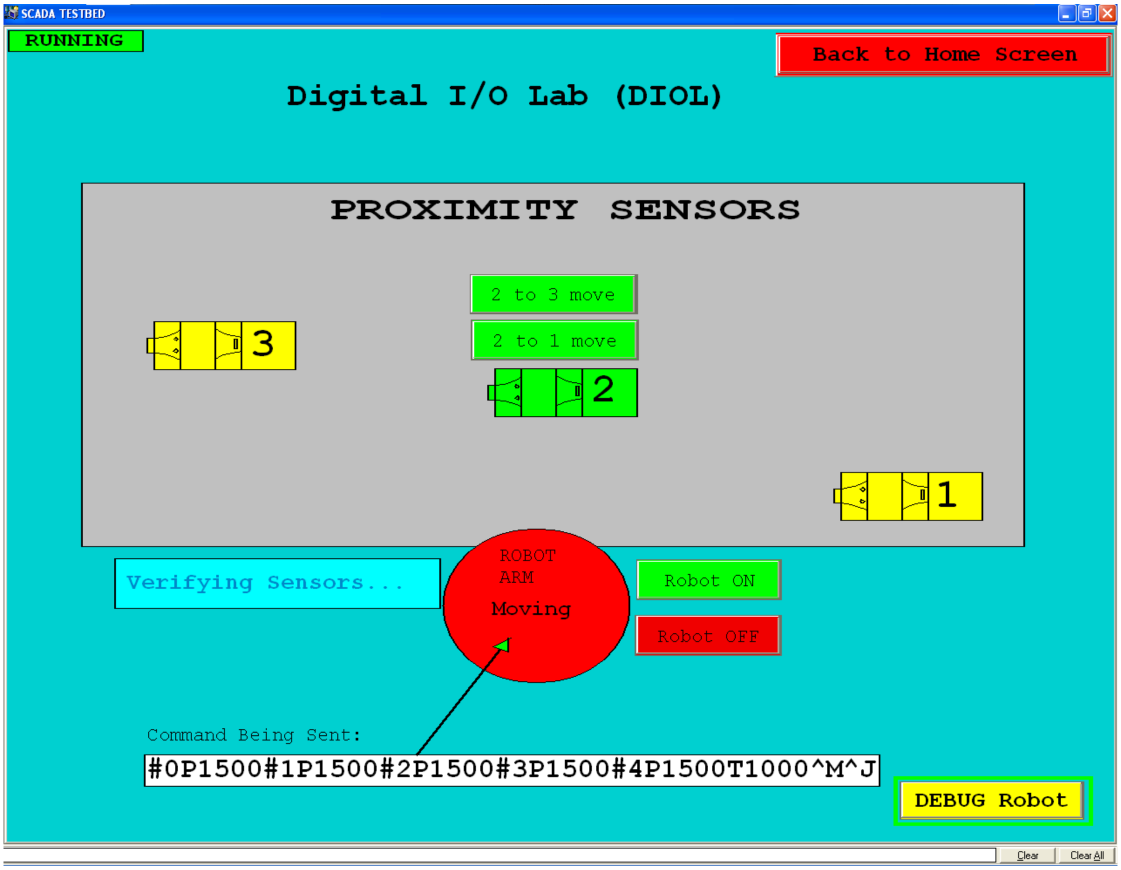
Step 5. HMI: click on the green “3 to 2 move” button

Observe: The robot arm picks up the object and moves it from sensor #3 to sensor #2.

Error Occured: If the robot arm drop or fails to deliver the object to sensor #2 then replace the object on the sensor by hand to continue the test.

Observe: HMI: commands to the robot are displayed and text “Sending Command” flashes as the robot moves. The robot arm graphic icon in the center of the screen turns red and the text “MOVING” flash.

Observe: HMI: After movement; a blue text box "Verifying sensors" appears, green move buttons appear above sensor #2 as choices for the robot arm. All other sensors are yellow and inactive (no button choices appear).



Verifying Sensors...

Command Being Sent:

```
#0P1500#1P1500#2P1500#3P1500#4P1500T1000^M^J
```

Observe: HMI: verification complete; the robot arm graphic changes color to green and the text "Moving" disappears.

Observe: Proximity Sensor: both the green and yellow LED indicators located on the sensor #3 are illuminated

Step 6. HMI: click on the green "2 to 1 move" button

Observe: The robot arm picks up the object and moves it from sensor #2 to sensor #1.

Error Occured: If the robot arm drop or fails to deliver the object to sensor #1 then replace the object on the sensor by hand to continue the test.

Observe: HMI: commands to the robot are displayed and text "Sending Command" flashes as the robot moves. The robot arm graphic icon in the center of the screen turns red and the text "MOVING" flash.

Observe: HMI: After movement; a blue text box "Verifying sensors" appears, green move buttons appear above sensor #1 as choices for the robot arm. All other sensors are yellow and inactive (no button choices appear).

Observe: Proximity Sensor: both the green and yellow LED indicators located on the sensor #1 are illuminated

Observe: HMI: verification complete; the robot arm graphic changes color to green and the text "Moving" disappears.

Step 7. HMI: click on the green "1 to 2 move" button

Observe: The robot arm picks up the object and moves it from sensor #1 to sensor #2.

Error Occured: If the robot arm drop or fails to deliver the object to sensor #2 then replace the object on the sensor by hand to continue the test.

Observe: HMI: commands to the robot are displayed and text "Sending Command" flashes as the robot moves. The robot arm graphic icon in the center of the screen turns red and the text "MOVING" flash.

Observe: HMI: After movement; a blue text box "Verifying sensors" appears, green move buttons appear above sensor #2 as choices for the robot arm. All other sensors are yellow and inactive (no button choices appear).

Observe: Proximity Sensor: both the green and yellow LED indicators located on the sensor #3 are illuminated

Observe: HMI: verification complete; the robot arm graphic changes color to green and the text "Moving" disappears.

Step 8. HMI: click on the green "2 to 3 move" button

Observe: The robot arm picks up the object and moves it from sensor #2 to sensor #3.

Error Occured: If the robot arm drop or fails to deliver the object to sensor #3 then replace the object on the sensor by hand to continue the test.

Observe: HMI: commands to the robot are displayed and text "Sending Command" flashes as the robot moves. The robot arm graphic icon in the center of the screen turns red and the text "MOVING" flash.

Observe: HMI: After movement; a blue text box "Verifying sensors" appears, green move buttons appear above sensor #3 as choices for the robot arm. All other sensors are yellow and inactive (no button choices appear).

Observe: HMI: verification complete; the robot arm graphic changes color to green and the text "Moving" disappears.

Step 9. HMI: click on the green "3 to 1 move" button

Observe: The robot arm picks up the object and moves it from sensor #3 to sensor #1.

Error Occured: If the robot arm drop or fails to deliver the object to sensor #1 then replace the object on the sensor by hand to continue the test.

Observe: HMI: commands to the robot are displayed and text "Sending Command" flashes as the robot moves. The robot arm graphic icon in the center of the screen turns red and the text "MOVING" flash.

Observe: HMI: After movement; a blue text box "Verifying sensors" appears, green move buttons appear above sensor #1 as choices for the robot arm. All other sensors are yellow and inactive (no button choices appear).

Observe: Proximity Sensor: both the green and yellow LED indicators located on the sensor #1 are illuminated

Observe: HMI: verification complete; the robot arm graphic changes color to green and the text "Moving" disappears.

Step 10. HMI: click on the green “1 to 3 move” button

Observe: The robot arm picks up the object and moves it from sensor #1 to sensor #3.

Error Occured: If the robot arm drop or fails to deliver the object to sensor #3 then replace the object on the sensor by hand to continue the test.

Observe: HMI: commands to the robot are displayed and text “Sending Command” flashes as the robot moves. The robot arm graphic icon in the center of the screen turns red and the text “MOVING” flash.

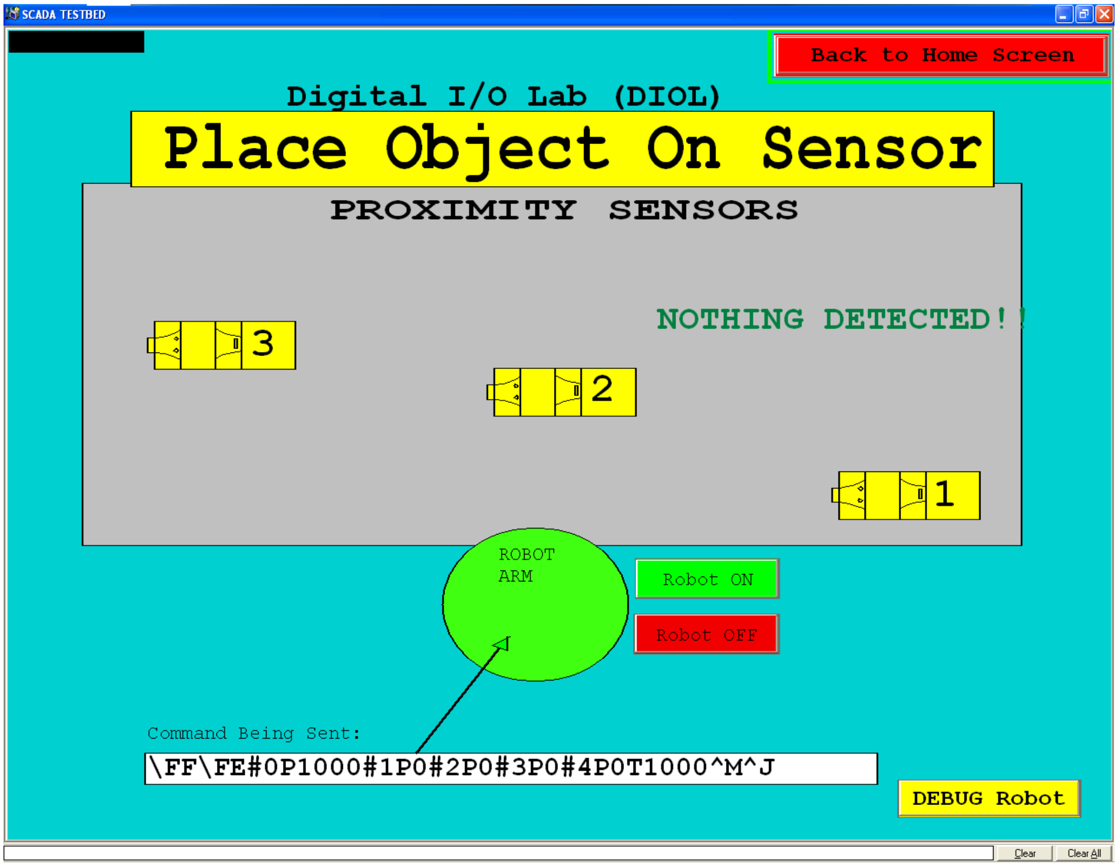
Observe: HMI: After movement; a blue text box “Verifying sensors” appears, green move buttons appear above sensor #3 as choices for the robot arm. All other sensors are yellow and inactive (no button choices appear).

Observe: Proximity Sensor: both the green and yellow LED indicators located on the sensor #3 are illuminated

Observe: HMI: verification complete; the robot arm graphic changes color to green and the text “Moving” disappears.

Step 11. Remove the object from sensor #3

Observe: HMI: A warning is displayed indicating no sensors are active.



Place Object On Sensor

NOTHING DETECTED!!

Command Being Sent: \FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J

Step 12. HMI: click on the red “Robot OFF” button

Observe: The robot arm goes limp

Observe: HMI: the command “\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J” is displayed in the box “Command Being Sent”.

Step 13. HMI: click the red “Back to Home Screen” button

Observe: The lab has been returned to the Ready State

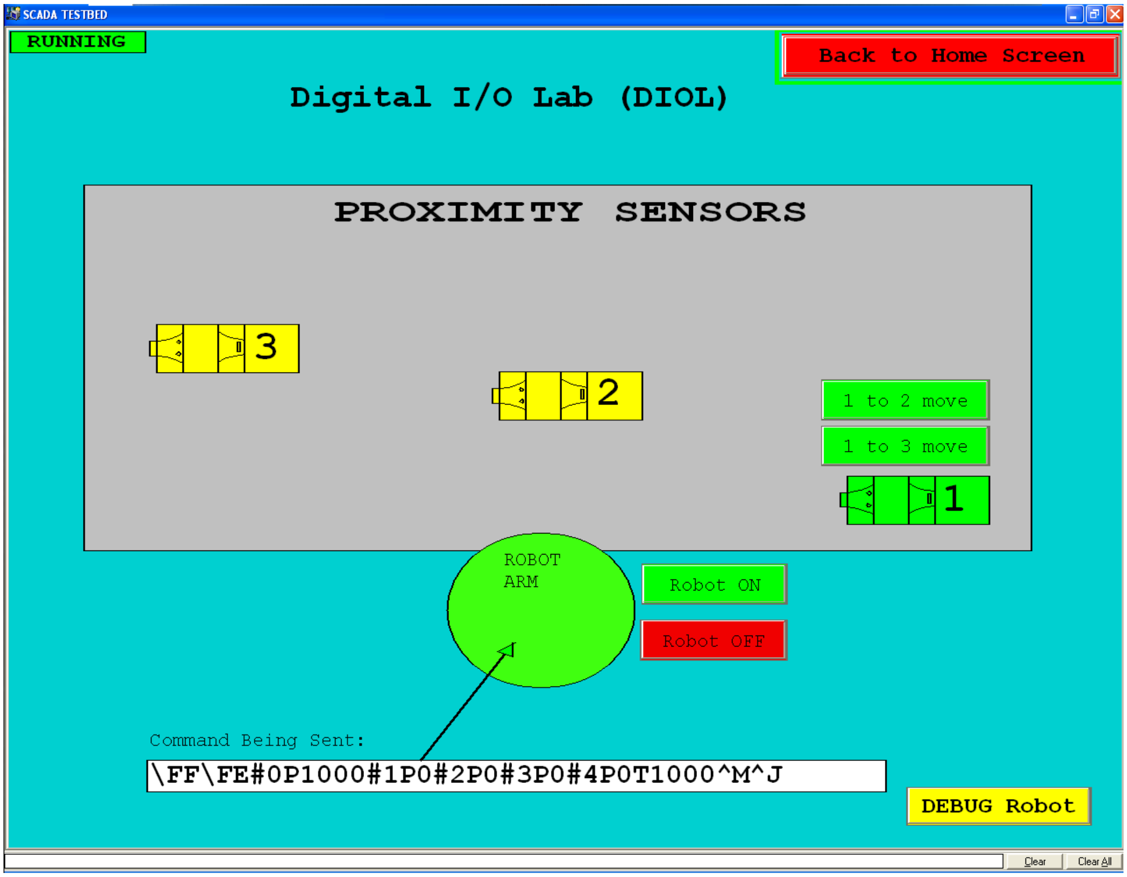
Observe: The test TF9 is complete

TEST TF10: Properly display proximity sensor status

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the blue “DIGITAL IO LAB” button.

Step 3. Place an object on sensor #1



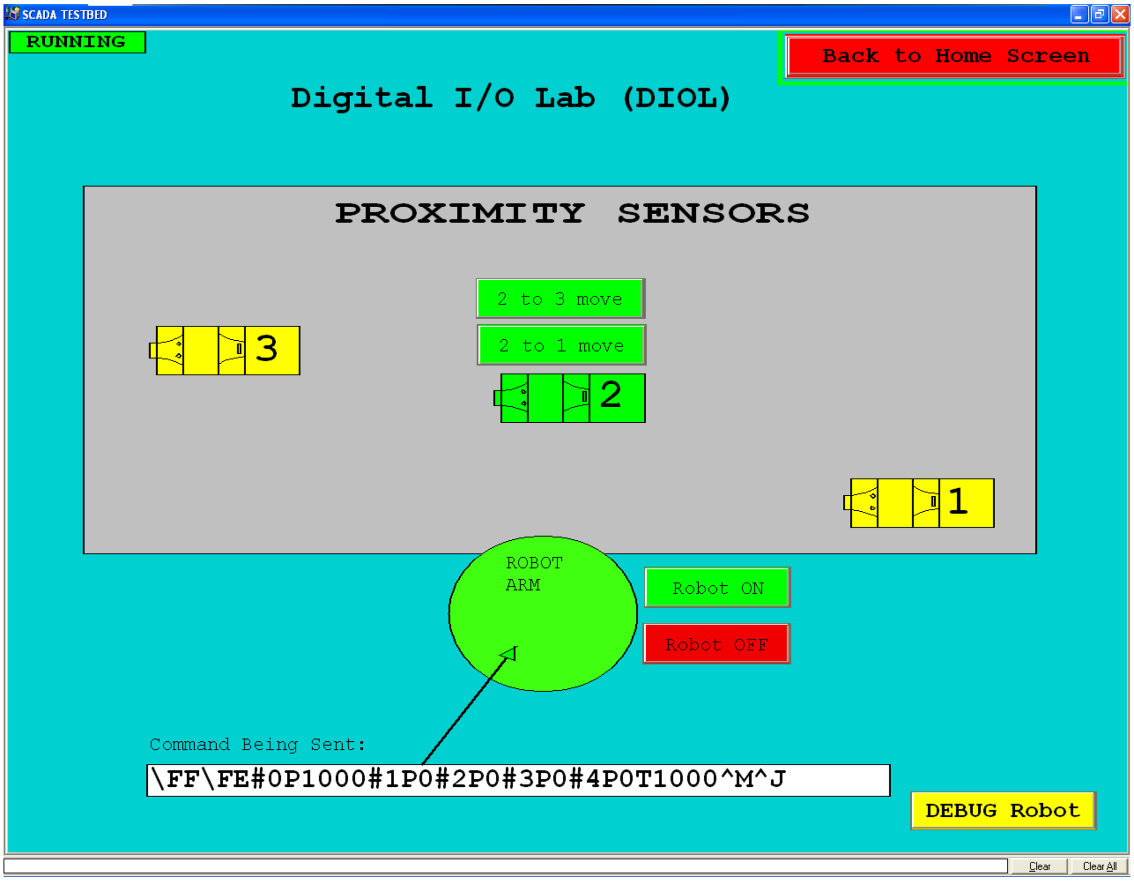
Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observe: HMI: sensor icon #1 is green. The green “1 to 2 move” and “1 to 3 move” buttons are visible

Observe: Proximity Sensor: both the green and yellow LED indicators located on the sensor #1 are illuminated

Step 4. Place an object on sensor #2



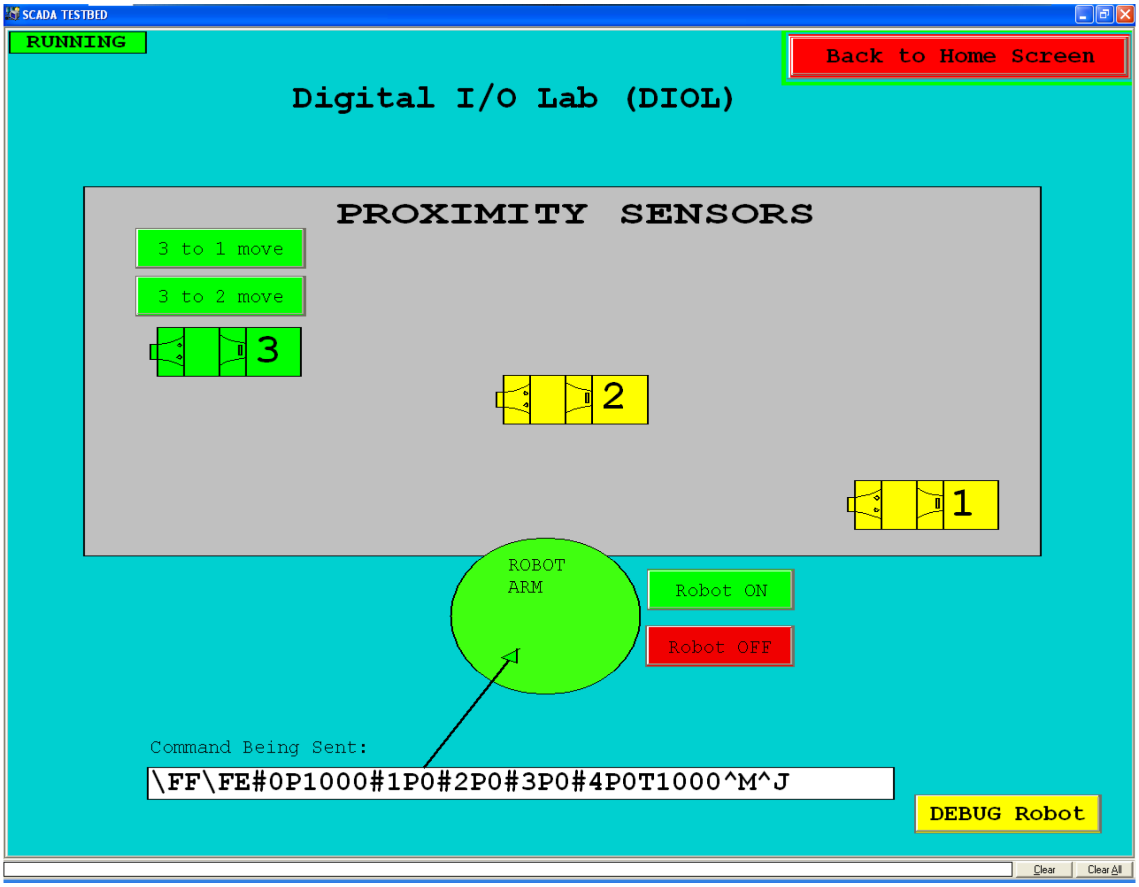
Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observe: HMI: sensor icon #2 is green. The green “2 to 3 move” and “2 to 1 move” robot command buttons are visible

Observe: Proximity Sensor: both the green and yellow LED indicators located on the sensor #2 are illuminated

Step 5. Place an object on sensor #3



Digital I/O Lab (DIOL)

PROXIMITY SENSORS

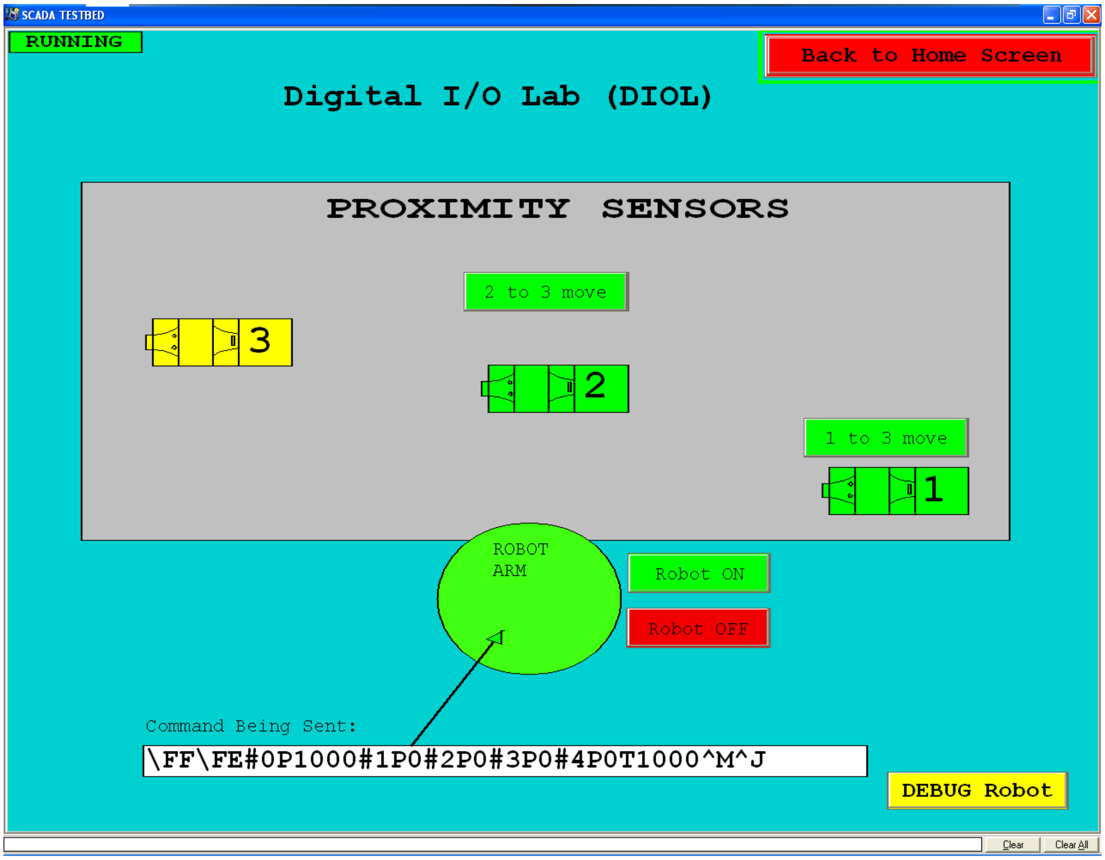
Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observe: HMI: sensor icon #3 is green. The green “3 to 1 move” and “3 to 2 move” robot command buttons are visible

Observe: Proximity Sensor: both the green and yellow LED indicators located on the sensor #3 are illuminated

Step 6. Place an object on sensors #1 and #2



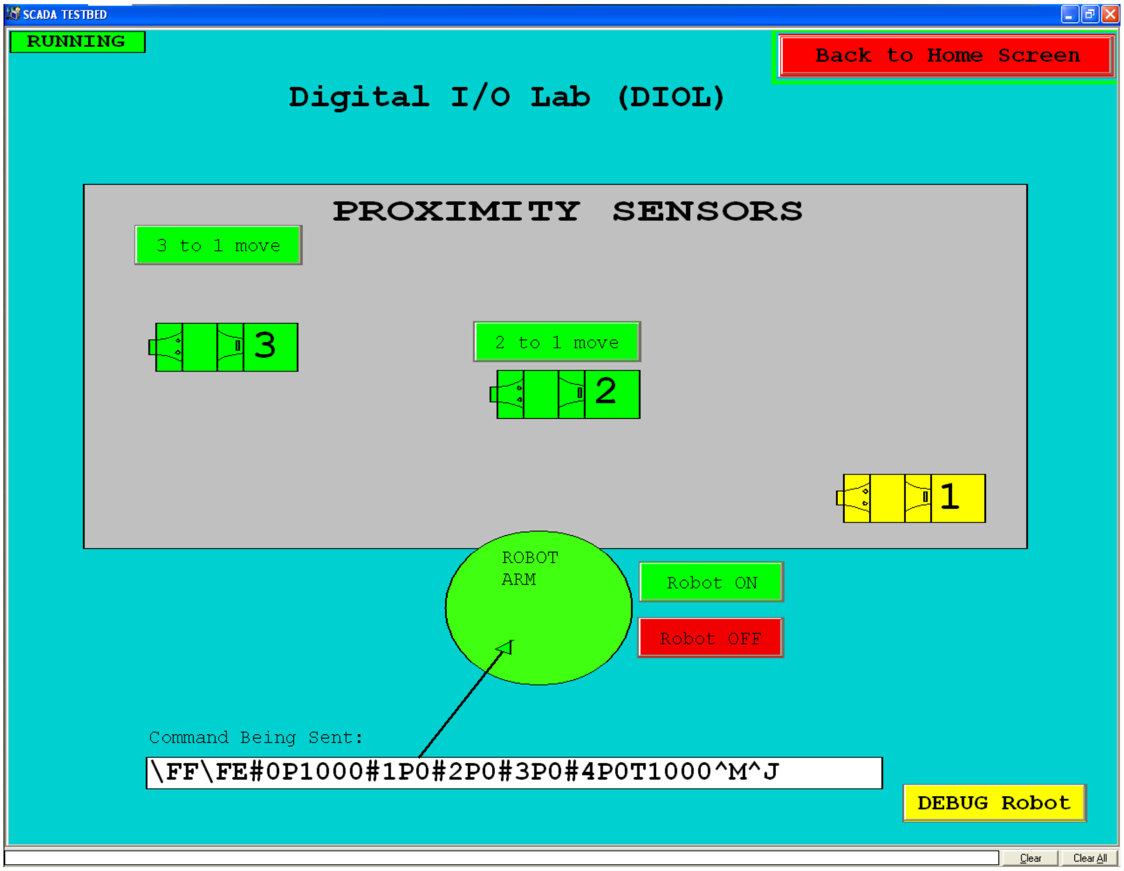
Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observe: HMI: sensor icon #1 and #2 are green. The green “1 to 3 move” and “2 to 3 move” robot command buttons are visible

Observe: Proximity Sensor: both the green and yellow LED indicators located on sensors #1 and #2 are illuminated

Step 7. Place an object on sensors #2 and #3



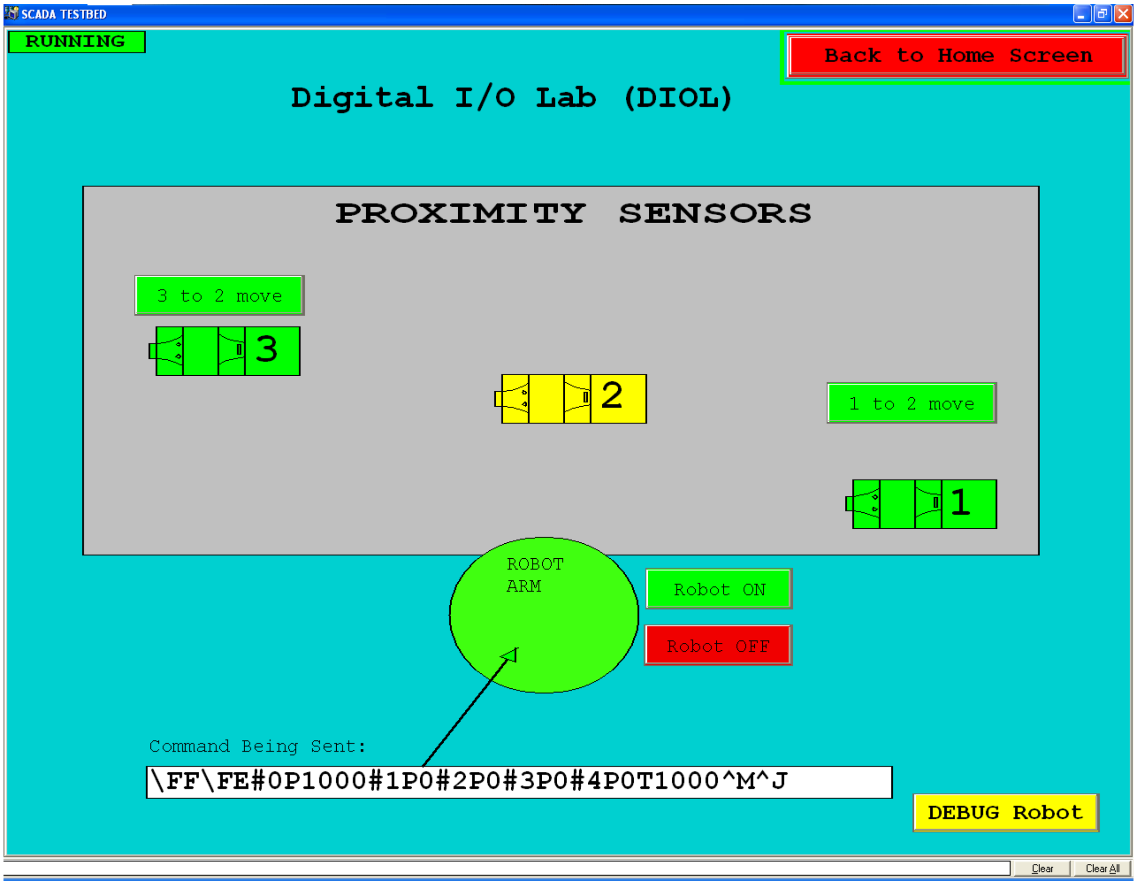
Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observe: HMI: sensor icon #2 and #3 are green. The green "2 to 1 move" and "3 to 1 move" robot command buttons are visible.

Observe: Proximity Sensor: both the green and yellow LED indicators located on sensors #2 and #3 are illuminated

Step 8. Place an object on sensors #1 and #3



Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observe: HMI: sensor icon #1 and #3 are green. The green "1 to 2 move" and "3 to 2 move" robot command buttons are visible.

Observe: Proximity Sensor: both the green and yellow LED indicators located on sensors #1and #3 are illuminated

Step 9. Remove all objects from the sensors.

Observe: HMI: all sensors turn yellow, a yellow warning sign "Place object on sensor" appears and the text "Nothing Detected" appears.

Step 10. HMI: click the red "Back to Home Screen" button

Observe: The lab has been returned to the Ready State

Observe: The test TF10 is complete

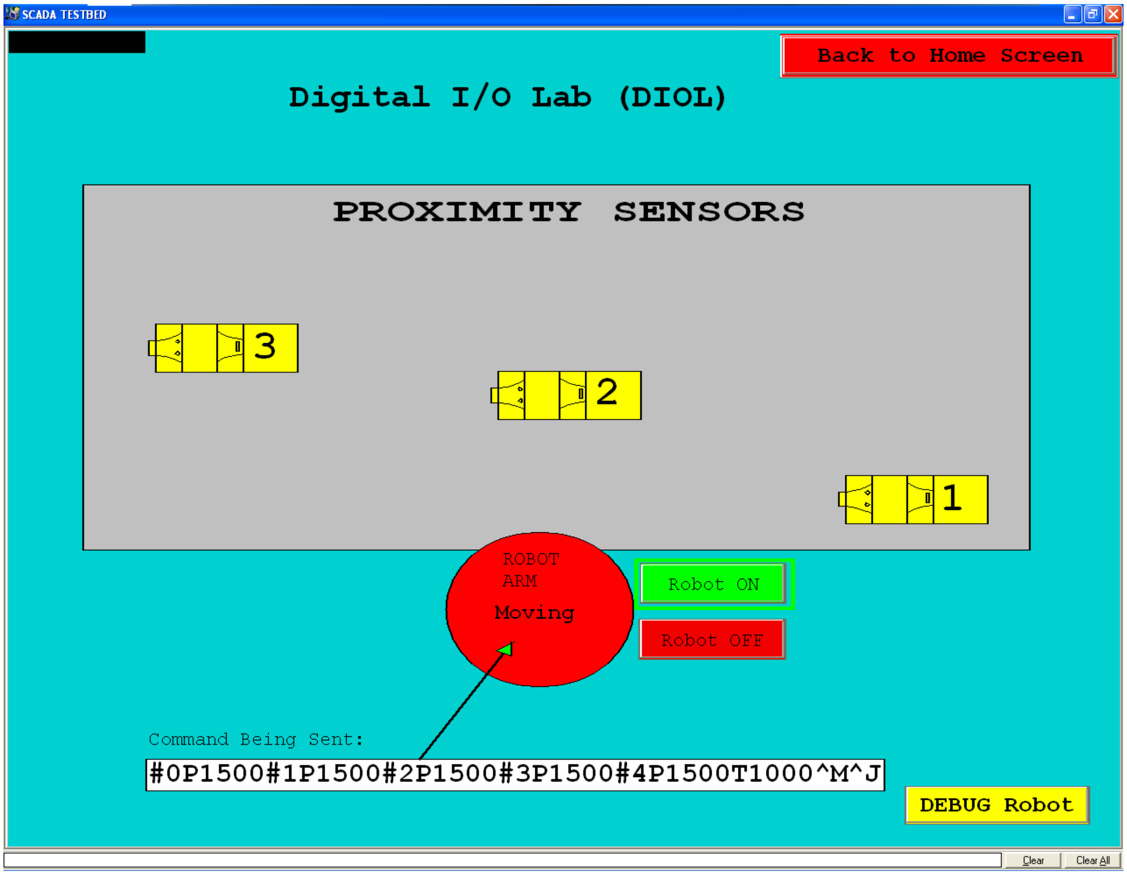
TEST TF11: Properly display robot arm status

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the blue "DIGITAL IO LAB" button.

Step 3. HMI: click the green "Robot ON" button

Observe: HMI: the center "ROBOT ARM" graphic turns red and the text "Moving" appear briefly.



Command Being Sent:

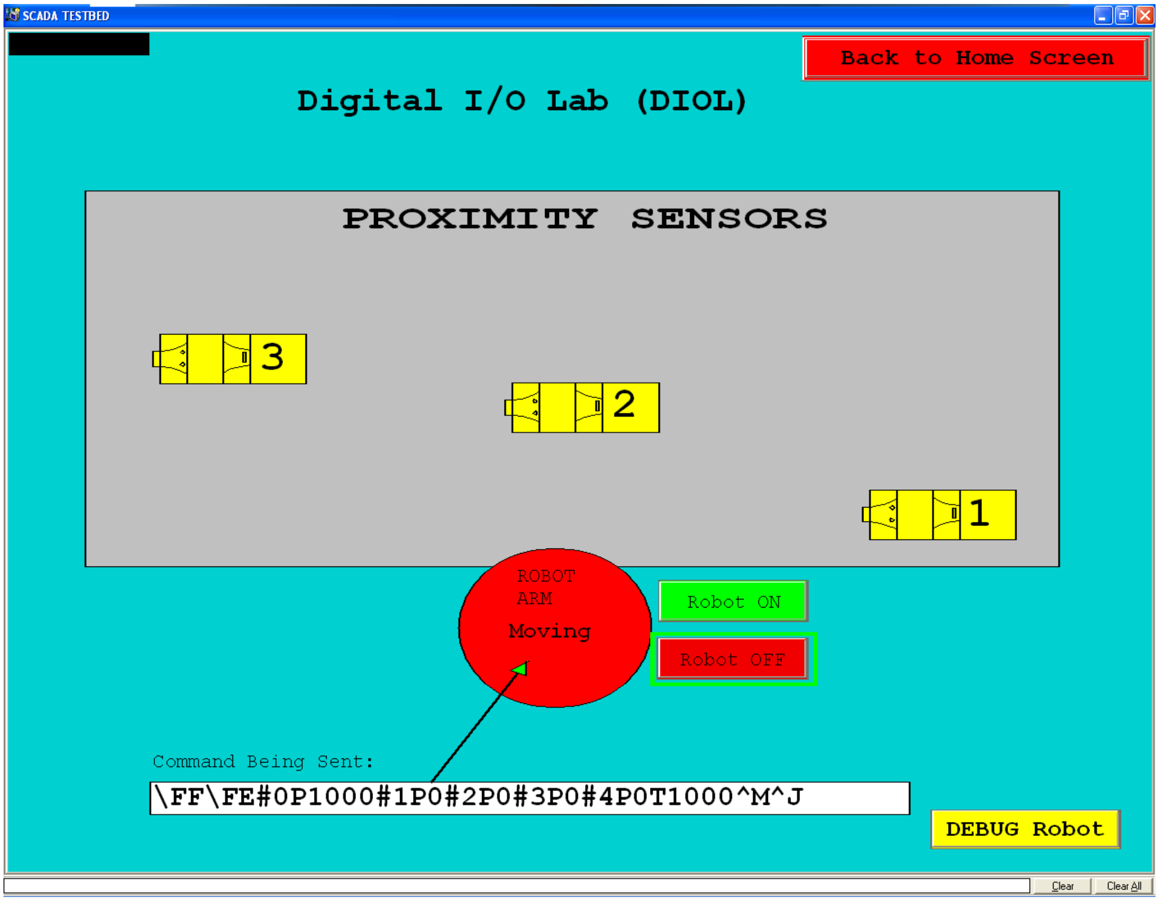
```
#0P1500#1P1500#2P1500#3P1500#4P1500T1000^M^J
```

Observe: The robot arm rises up to the "stow" position and remains there

Observe: HMI: the center "ROBOT ARM" graphic turns green. No "Moving" text is visible. (signifying the robot is ready for another command)

Step 4. HMI: click the red “Robot OFF” button

Observe: HMI: the center “ROBOT ARM” graphic turns red and the text “Moving” appear briefly.



[redacted]

Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observe: HMI: the center “ROBOT ARM” graphic turns red and the text “Moving” appear briefly.

Observe: The robot arm falls to the “off” position and remains there

Observe: HMI: the center “ROBOT ARM” graphic turns green. No “Moving” text is visible. (signifying the robot is ready for another command)

Step 5. HMI: click the red “Back to Home Screen” button

Observe: The lab has been returned to the Ready State

Observe: The test TF11 is complete

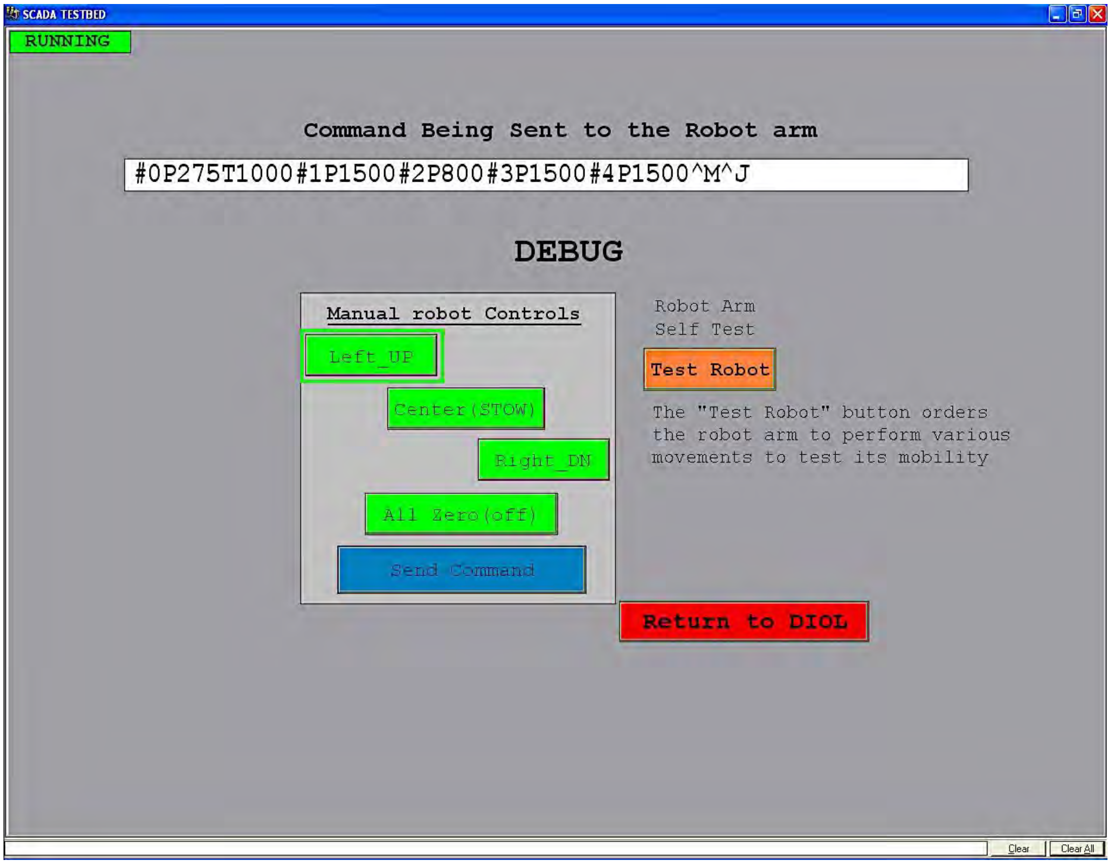
TEST TF12: Properly display robot arm commands as they are issued

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the blue “DIGITAL IO LAB” button.

Step 3. HMI: click on the yellow “DEBUG Robot” button

Observe: HMI: the Debug screen appears



Command Being Sent to the Robot arm

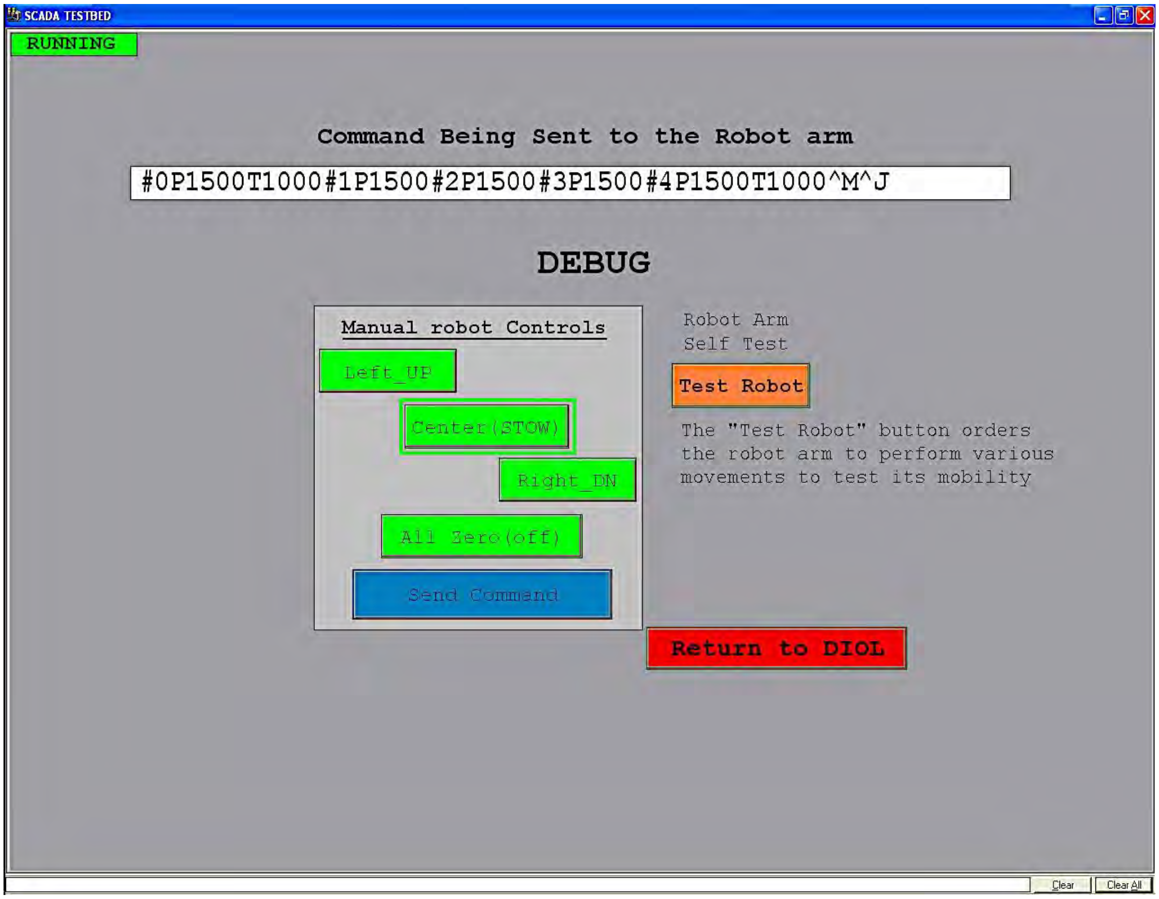
#0P275T1000#1P1500#2P800#3P1500#4P1500^M^J

The "Test Robot" button orders the robot arm to perform various movements to test its mobility

Step 4. HMI: click on the green “Left_UP” button.

Observe: The “Command Being Sent to the Robot arm” text box displays the following command:

“#0P275T1000#1P1500#2P800#3P1500#4P1500^M^J”



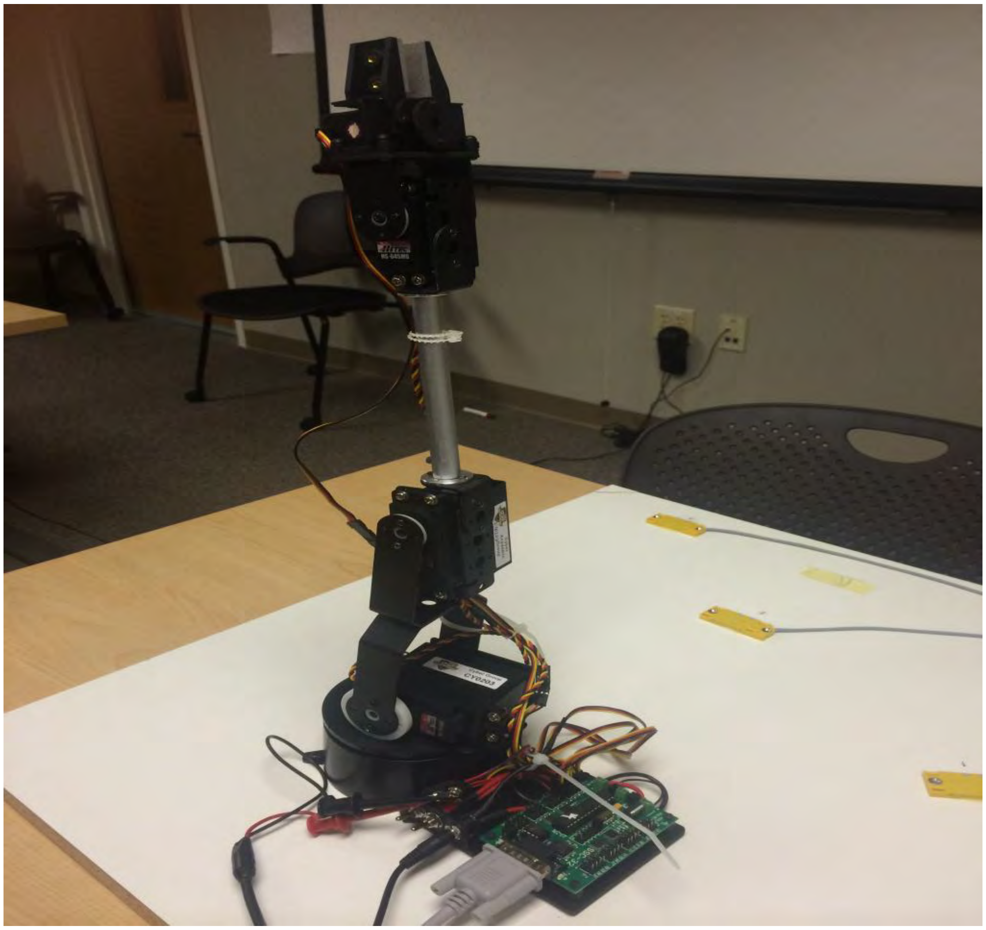
Command Being Sent to the Robot arm

```
#0P1500T1000#1P1500#2P1500#3P1500#4P1500T1000^M^J
```

The "Test Robot" button orders the robot arm to perform various movements to test its mobility

Step 5. HMI: click and hold the blue "Send Command" button for 1-2 seconds and then release.

Observe: The robot arm moves to the left and up to the position ordered by the command "#0P275T1000#1P1500#2P800#3P1500#4P1500^M^J"



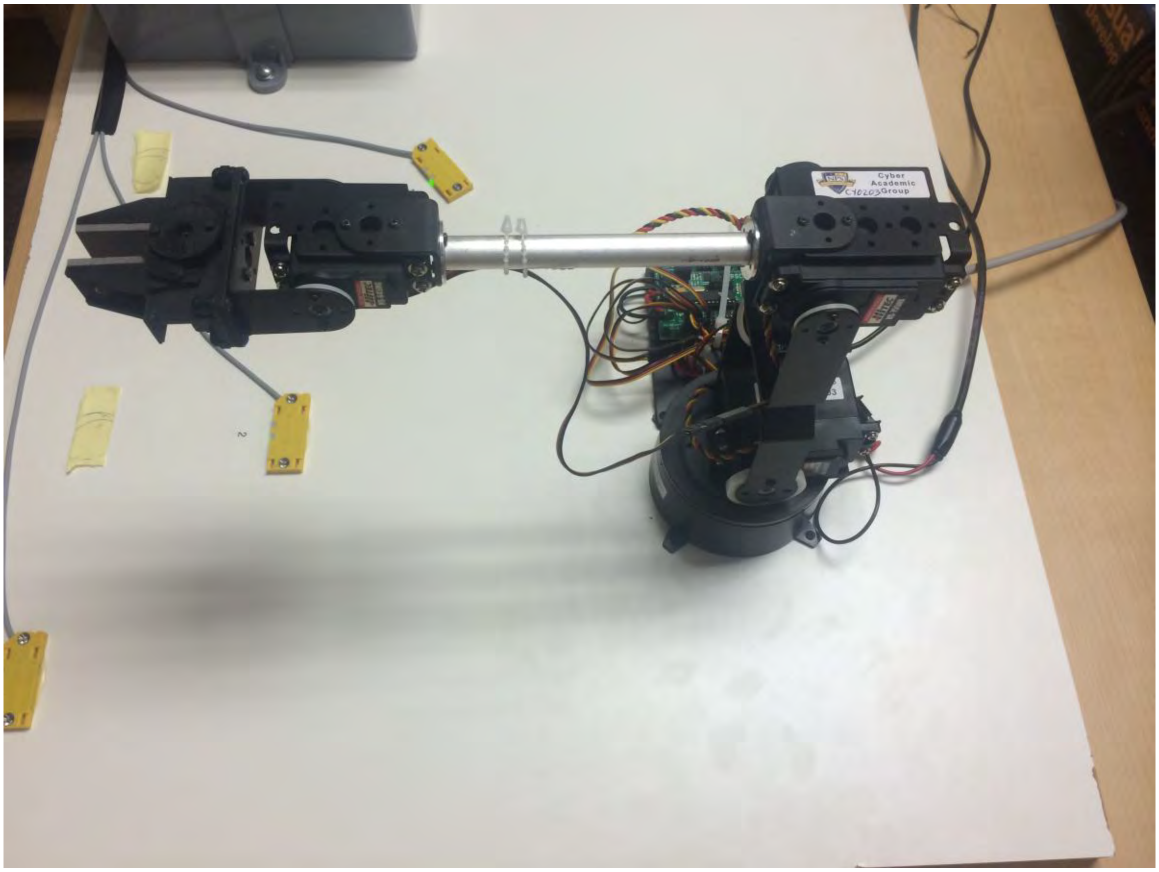
Step 6. HMI: click on the green “Center (STOW)” button

Observe: The “Command Being Sent to the Robot arm” text box displays the following command:

“#0P1500T1000#1P1500#2P1500#3P1500#4P1500T1000^M^J”

Step 7. HMI: click on the blue “Send Command” button

Observe: The robot arm returns to the “Stow” position. The STOW position is the default position of the robot arm between movements or commands.



Step 8. HMI: click on the red “Return to DIOL” button

Observe: HMI: the DIOL screen appears

Step 9. HMI: click on the red “Robot OFF” button

Observe: The robot arm falls to the “off” position and remains there

Observe: HMI: the center “ROBOT ARM” graphic eventually turns green. No “Moving” text is visible. (signifying the robot is ready for another command)

Step 10. HMI: click the red “Back to Home Screen” button

Observe: The lab has been returned to the Ready State

Observe: The test TF12 is complete

EXCEPTION TESTING

The following tests will exercise the controls put in place to catch anticipated exceptions in the operation of the MCS testbed. The scenarios demonstrate the implemented solutions to each of the exceptions.

TEST TE1: Tank level alarm value not within acceptable range or format.

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the green “ANALOG IO LAB” button.

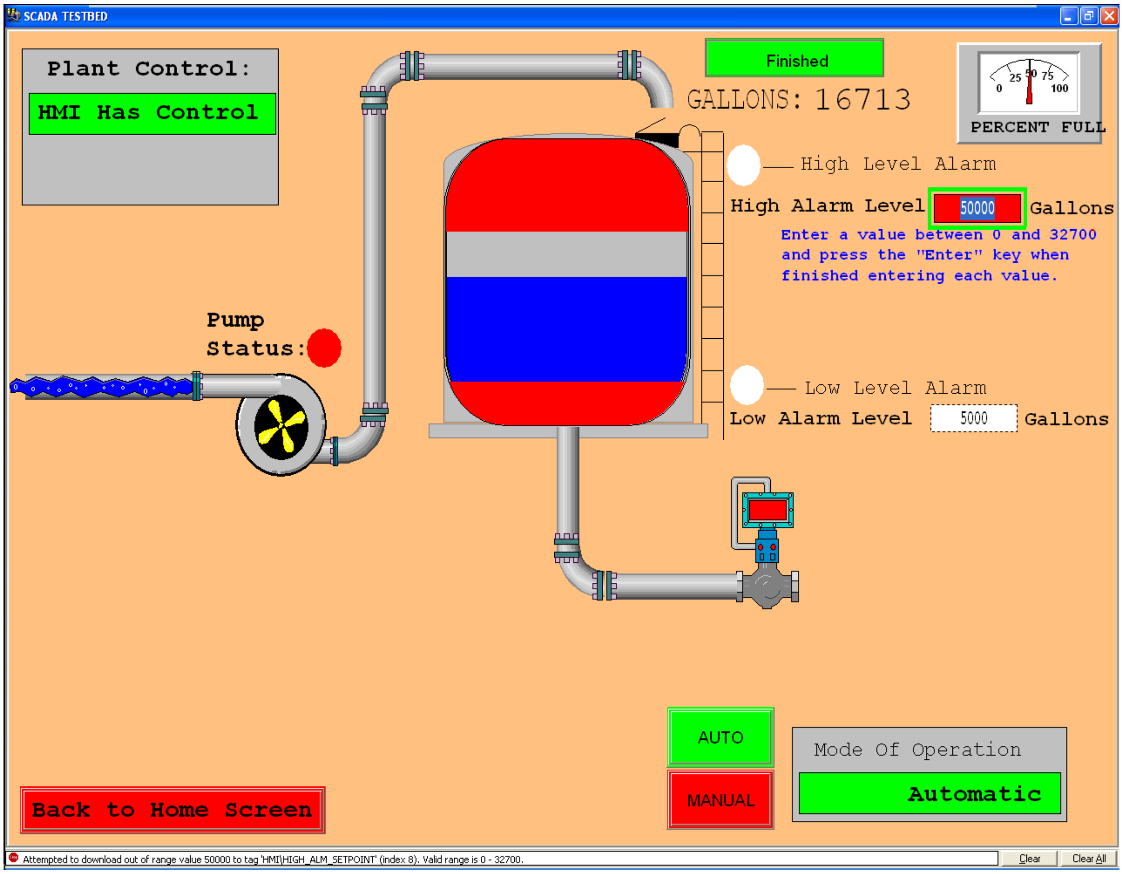
Step 3. HMI: click on the red “MANUAL” button and ensure the Mode of Operation is “Manual”

Step 4. HMI: click the red “Adjust Alarm Setpoints” button

Observation: HMI: the “High Alarm Level” and “Low Alarm Level” entry boxes appear and the graphical alarm levels appear in red inside the fluid tank

Step 5. HMI: enter “50000” gallons in the “High Alarm Level” entry box and hit Enter.

Observation: HMI: the “High Alarm Level” entry box turns red and the proposed new high alarm level is not accepted. Additionally an error will be displayed at the bottom of the screen indicating that an out of range value was entered.



HMI Has Control

GALLONS: 16713

High Alarm Level 50000 Gallons

Enter a value between 0 and 32700 and press the "Enter" key when finished entering each value.

Low Alarm Level 5000 Gallons

Mode Of Operation: Automatic

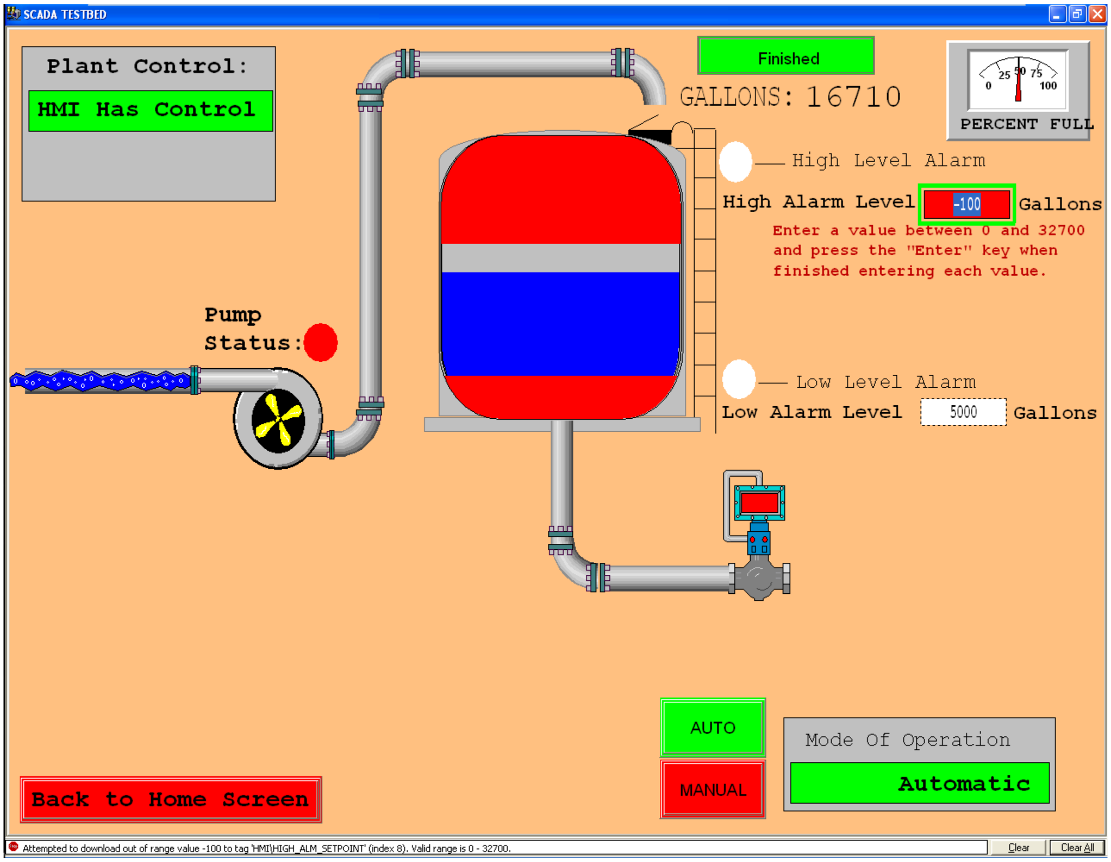
```
Attempted to download out of range value 50000 to tag 'HMI\HIGH_ALM_SETPOINT' (index 8). Valid range is 0 - 32700.
```

Step 6. HMI: enter “20000” gallons in the “High Alarm Level” entry box and hit Enter.

Observation: HMI: the “High Alarm Level” entry box turns white and the proposed value is accepted, the red graphic inside the tank representing the high alarm level changes to reflect 20000 gallons.

Step 7. HMI: enter “-100” gallons in the “High Alarm Level” entry box and hit Enter

Observation: HMI: the “High Alarm Level” entry box turns red and the proposed new alarm level is not accepted. Additionally an error will be displayed at the bottom of the screen indicating that an out of range value was entered.



Plant Control:
HMI Has Control

GALLONS: 16710

High Alarm Level -100 Gallons

Enter a value between 0 and 32700 and press the "Enter" key when finished entering each value.

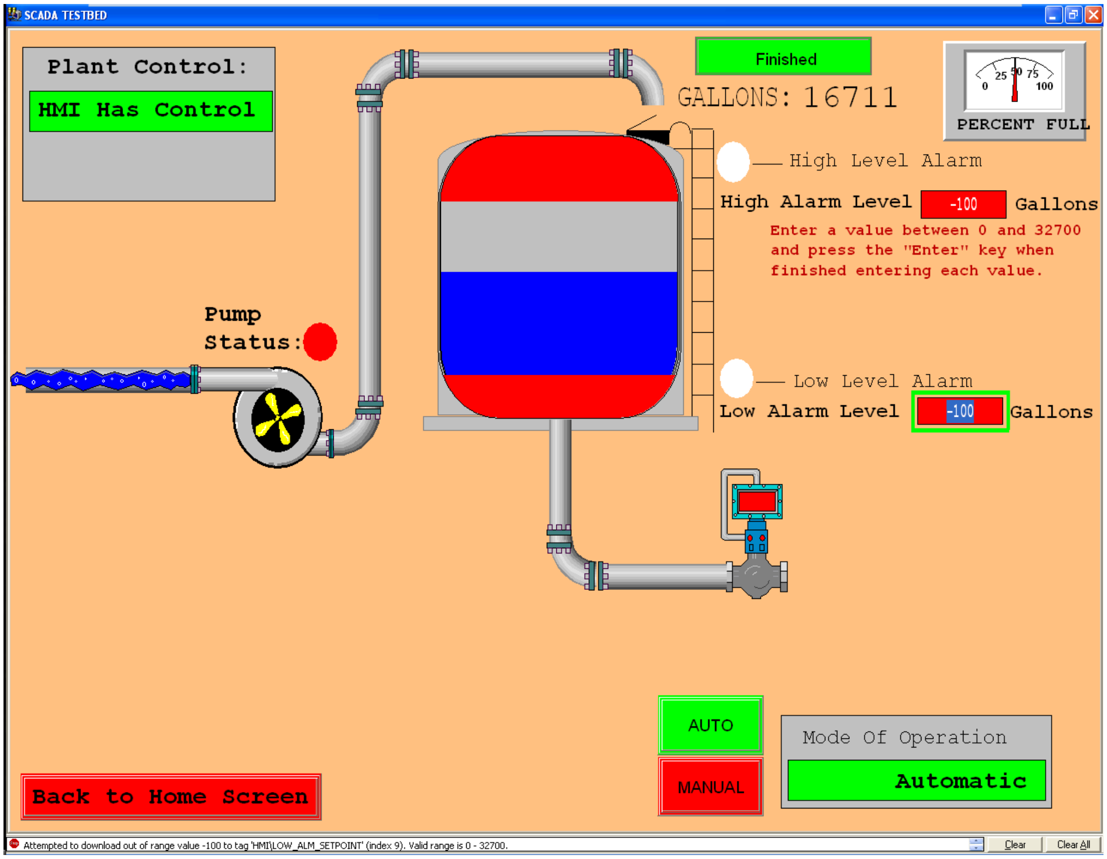
Low Alarm Level 5000 Gallons

Mode Of Operation
Automatic

```
Attempted to download out of range value -100 to tag 'HMI\HIGH_ALM_SETPOINT' (index 8). Valid range is 0 - 32700.
```

Step 8. HMI: enter "-100" gallons in the "Low Alarm Level" entry box and hit Enter

Observation: HMI: the "Low Alarm Level" entry box turns red and the proposed new alarm level is not accepted. Additionally an error will be displayed at the bottom of the screen indicating that an out of range value was entered.



Enter a value between 0 and 32700 and press the "Enter" key when finished entering each value.

```
Attempted to download out of range value -100 to tag 'HMI\LOW_ALM_SETPOINT' (index 9). Valid range is 0 - 32700.
```

Step 9. HMI: enter "5000" gallons in the "Low Alarm Level" entry box and enter "22000" gallons in the "High Alarm Level" and hit Enter.

Observation: HMI: the "High Alarm Level" and "Low Alarm Level" entry boxes turn white and the proposed values are accepted, the red graphic inside the tank representing the high alarm level changes to reflect 25000 and 5000 gallons respectively.

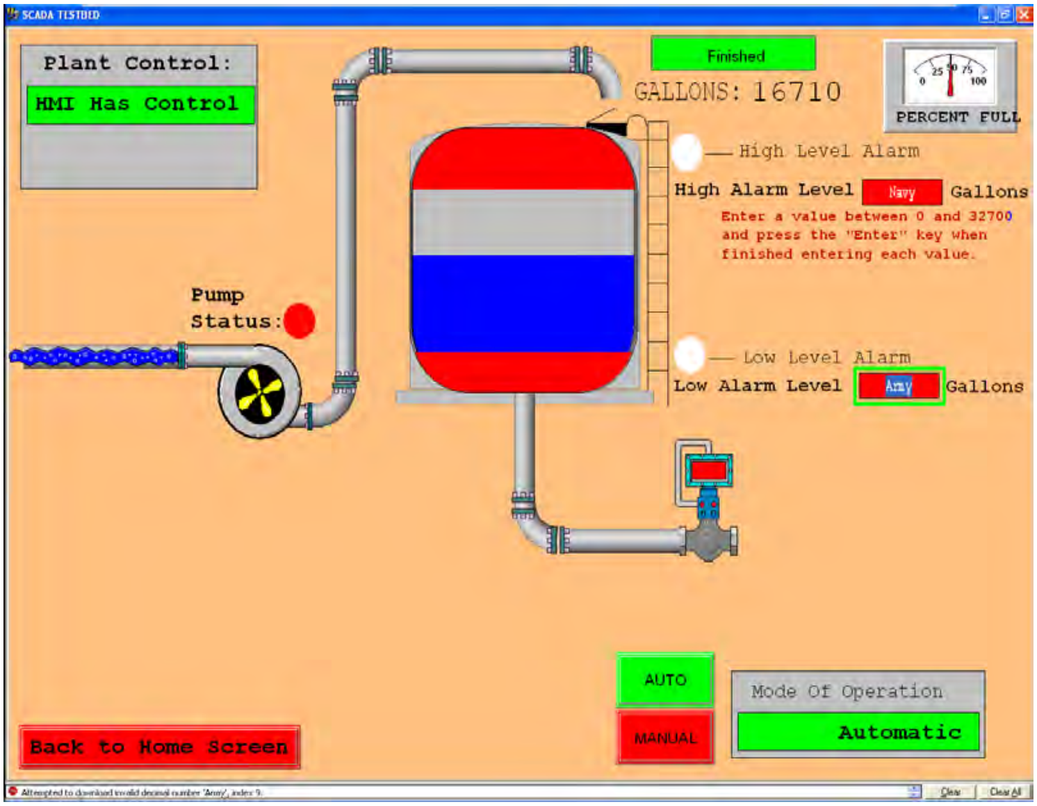
Step 10. HMI: click on the "Clear All" button in the lower right corner of the screen to clear all the errors produced.

Observation: HMI: The error screen to the left of the button will clear

Step 11. HMI: enter "Navy" gallons in the "High Alarm Level" entry box and hit Enter.

Observation: HMI: the "High Alarm Level" entry box turns red and the proposed new alarm level is not accepted. Additionally an error will be displayed at the bottom of the screen indicating that an invalid decimal number was entered

Step 12. HMI: enter “Army” gallons in the “Low Alarm Level” entry box and hit Enter



Enter a value between 0 and 32700 and press the "Enter" key when finished entering each value.

Attempted to download invalid decimal number 'Army', index 9.

Observation: HMI: the “Low Alarm Level” entry box turns red and the proposed new alarm level is not accepted. Additionally an error will be displayed at the bottom of the screen indicating that an invalid decimal number was entered

Step 13. HMI: enter “5000” gallons in the “Low Alarm Level” entry box and enter “25000” gallons in the “High Alarm Level” and hit Enter.

Observation: HMI: the “High Alarm Level” and “Low Alarm Level” entry boxes turn white and the proposed values are accepted, the red graphic inside the tank representing the high alarm level changes to reflect 25000 and 5000 gallons respectively.

Step 14. HMI: click on the “Clear All” button in the lower right corner of the screen to clear all the errors produced.

Observation: HMI: The error screen to the left of the button will clear

Step 15. HMI: click the green “Finished” button

Observation: HMI: options to enter High and Low alarm levels disappear.

Step 16. HMI: click the red “Back to Home Screen” button

Observation: The lab has been returned to the Ready State

Observation: The test TE1 is complete

TEST TE2: Illogical tank level alarm values (Low > High desired setpoint value)

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

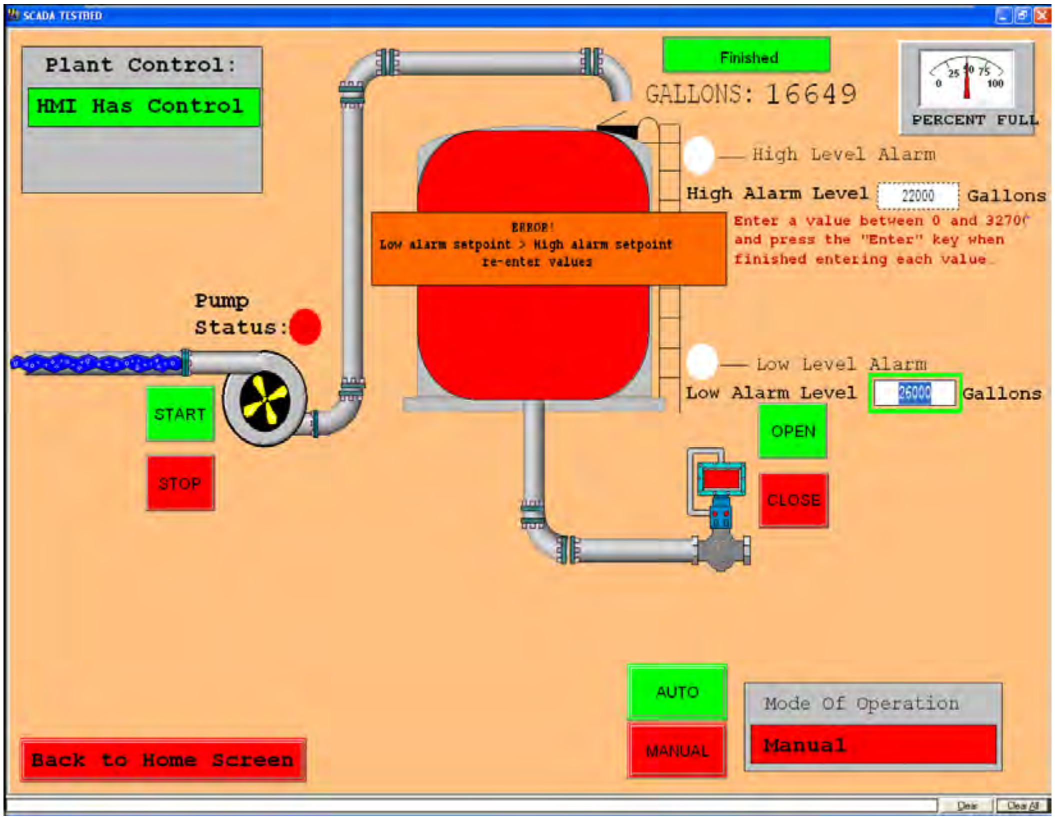
Step 2. HMI: click on the green “ANALOG IO LAB” button.

Step 3. HMI: click on the red “MANUAL” button and ensure the Mode of Operation is “Manual”

Step 4. HMI: click the red “Adjust Alarm Setpoints” button

Step 5. HMI: click on the “Low Alarm Level” text box and enter “26000” gallons and hit enter.

Observation: HMI: a text box with an error will appear over the tank indicating that the value entered is greater than the High Level Alarm setpoint and to re-enter the value.



Plant Control: HMI Has Control

GALLONS: 16649

High Alarm Level 22000 Gallons

Enter a value between 0 and 3270(and press the "Enter" key when finished entering each value.

ERROR!
Low alarm setpoint > High alarm setpoint
re-enter values

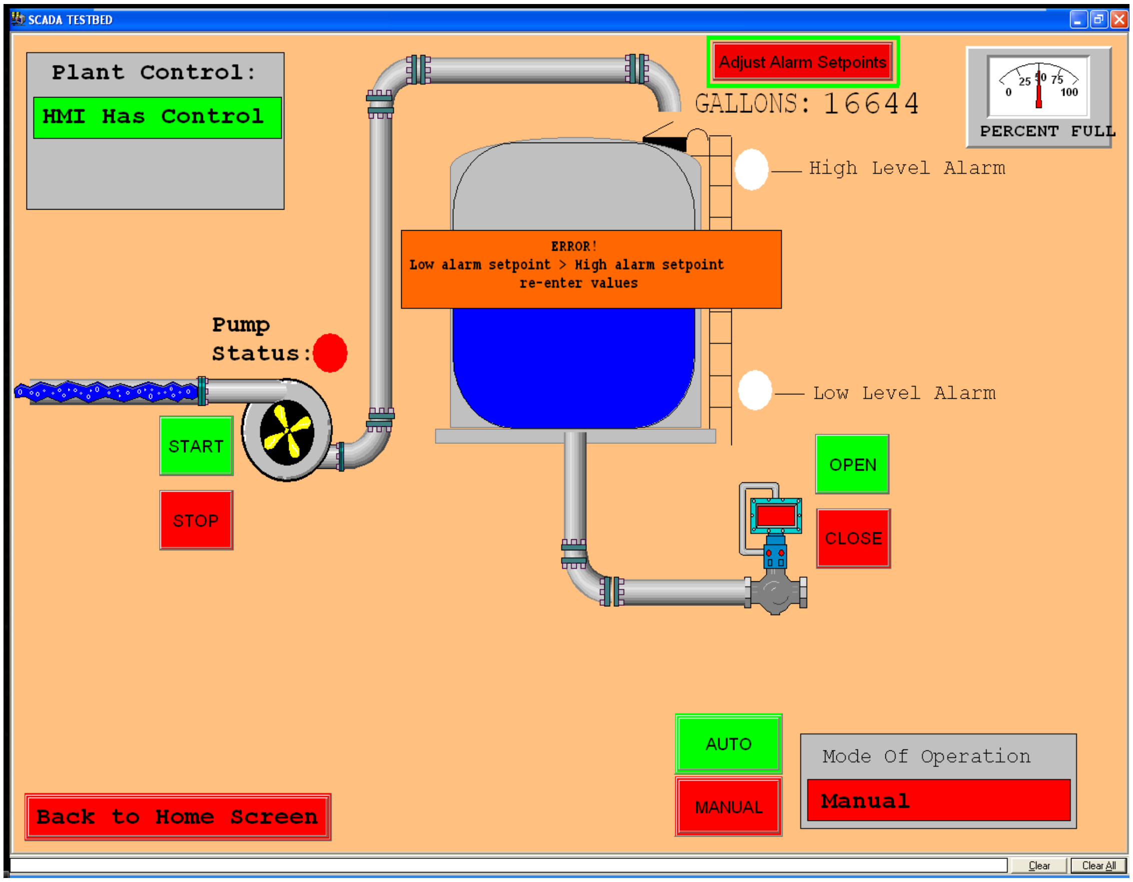
Low Alarm Level 26000 Gallons

Mode Of Operation: Manual

Observation: HOS: the red High or Low level alarm LED’s remain extinguished.

Step 6. HMI: click the green “Finished” button

Observation: HMI: options to enter High and Low alarm levels disappear but the error text and box remain.



SCADA TESTBED

Plant Control:

HMI Has Control

GALLONS: 16644

ERROR!
Low alarm setpoint > High alarm setpoint
re-enter values

Mode Of Operation

Manual

Observation: HOS: the red High or Low level alarm LED’s remain extinguished.

Step 7. HMI: click the red “Adjust Alarm Setpoints” button again.

Step 8. HMI: click on the “Low Alarm Level” text box and enter “5000” gallons and hit enter.

Observation: HMI: illogical setpoints Error clears and system is normal.

Step 9. HMI: click on the “High Alarm Level” text box and enter “4900” gallons and hit enter.

Observation: HMI: a text box with an error will appear over the tank indicating that the value entered is greater than the High Level Alarm setpoint and to re-enter the value.

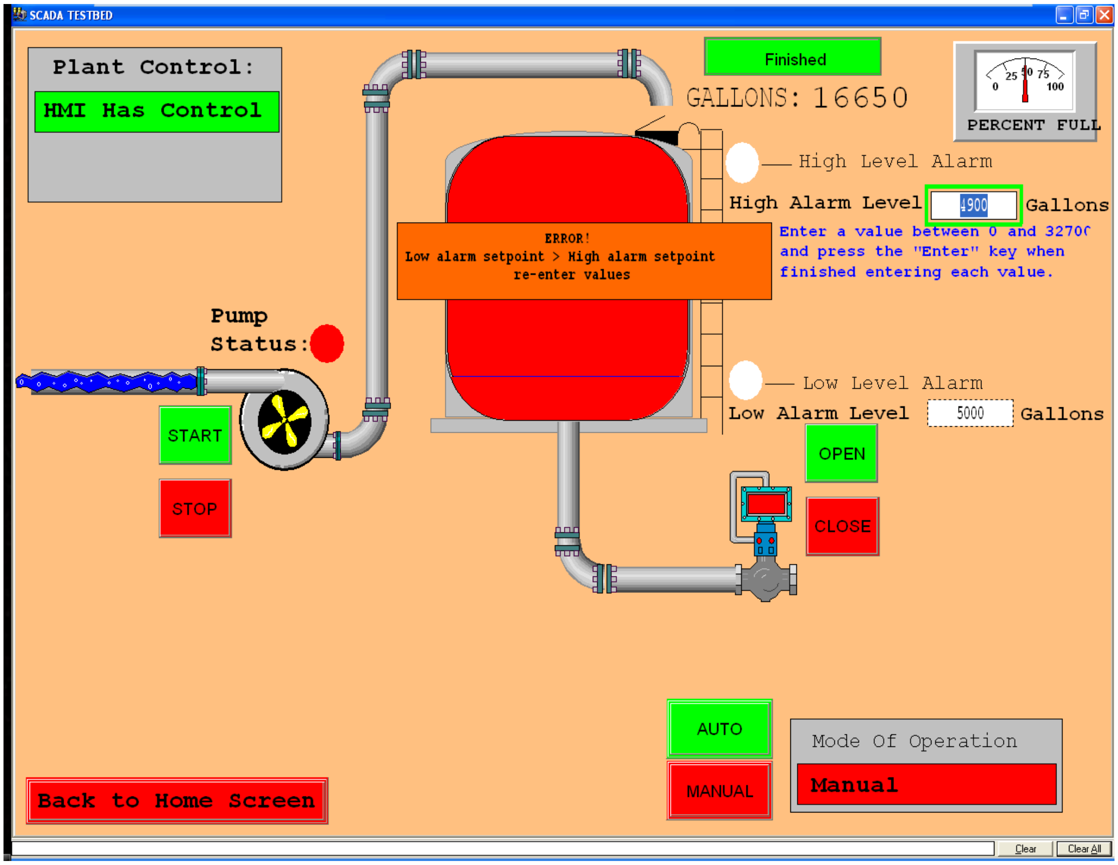
Observation: HOS: the red High or Low level alarm LED's remain extinguished.

Step 10. HMI: click on the green "Finished" button

Observation: HMI: the error remains overtop of the tank graphic

Step 11. HMI: click on the red "Adjust Alarm Setpoint" button

Observation: HMI: the setpoint boxes appear again



Plant Control: HMI Has Control

GALLONS: 16650

High Alarm Level 4900 Gallons

Enter a value between 0 and 3270(
and press the "Enter" key when
finished entering each value.

ERROR!
Low alarm setpoint > High alarm setpoint
re-enter values

Low Alarm Level 5000 Gallons

Mode Of Operation Manual

Step 12. HMI: click on the "High Alarm Level" text box and enter "22000" gallons and hit enter.

Observation: HMI: illogical setpoints error clears and system returns to normal.

Step 13. HMI: click the green “Finished” button

Step 14. HMI: click the red “Back to Home Screen” button

Observation: The lab has been returned to the Ready State

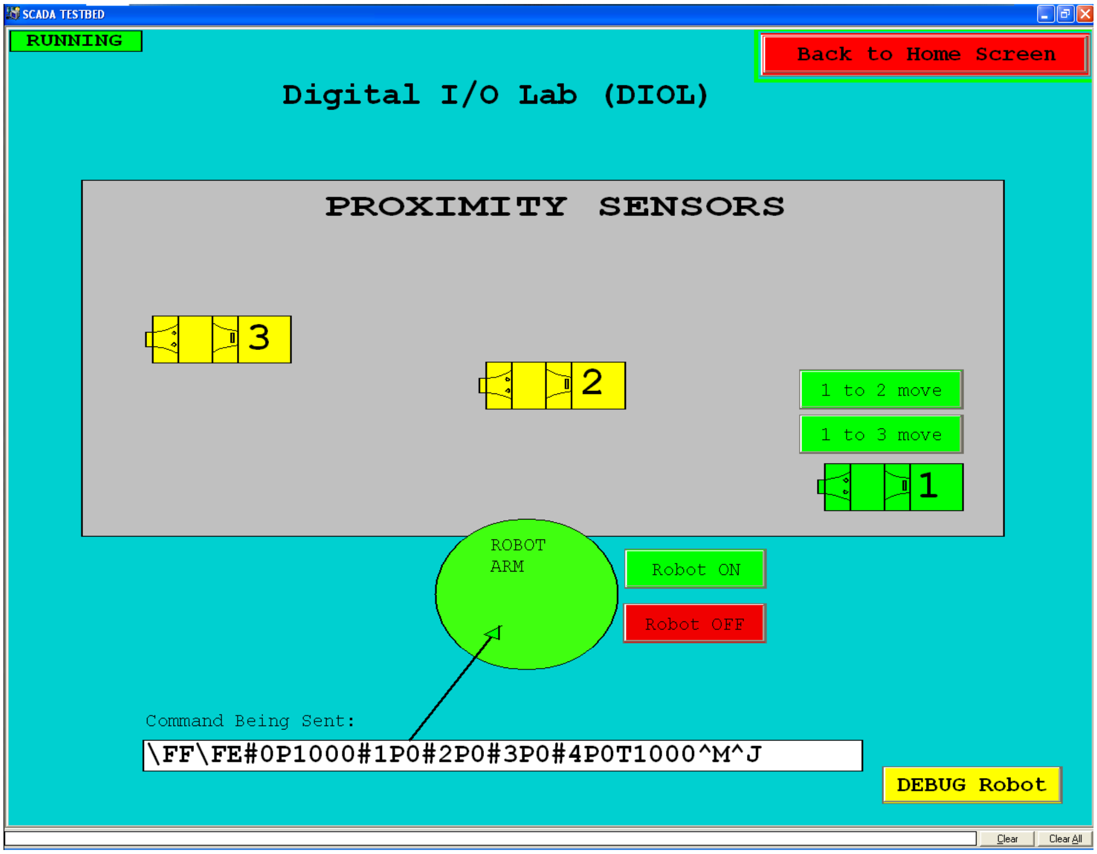
Observation: The test TE2 is complete

TEST TE3: Robot arm fails to move or deliver object to sensor

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the blue "DIGITAL IO LAB" button.

Step 3. Place the barrel object on proximity sensor # 1.



Command Being Sent:

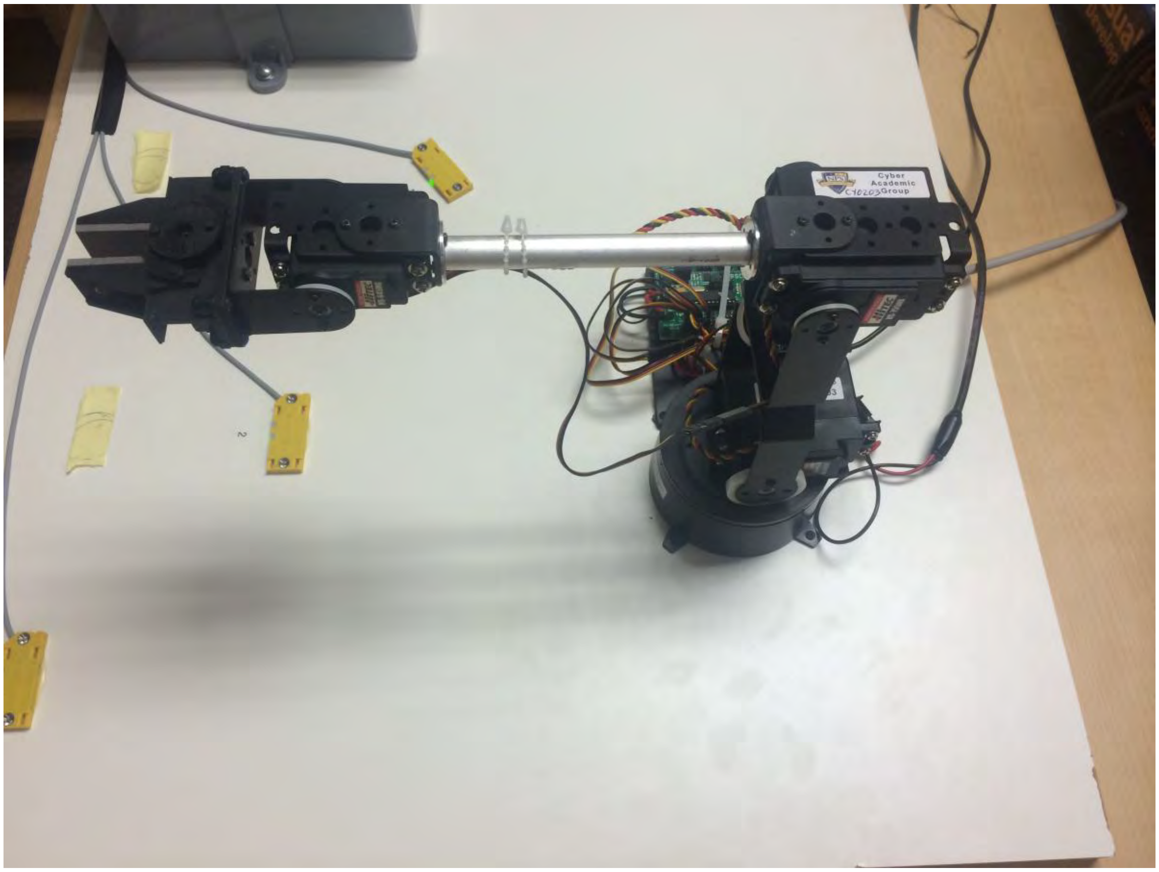
```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observation: HMI: sensor icon #1 is green. The green "1 to 2 move" and "1 to 3 move" buttons are visible

Observation: Proximity Sensor: both the green and yellow LED indicators located on the sensor #1 are illuminated

Step 4. HMI: click the green "Robot ON" button

Observation: The robot arm moves to the "stow" position

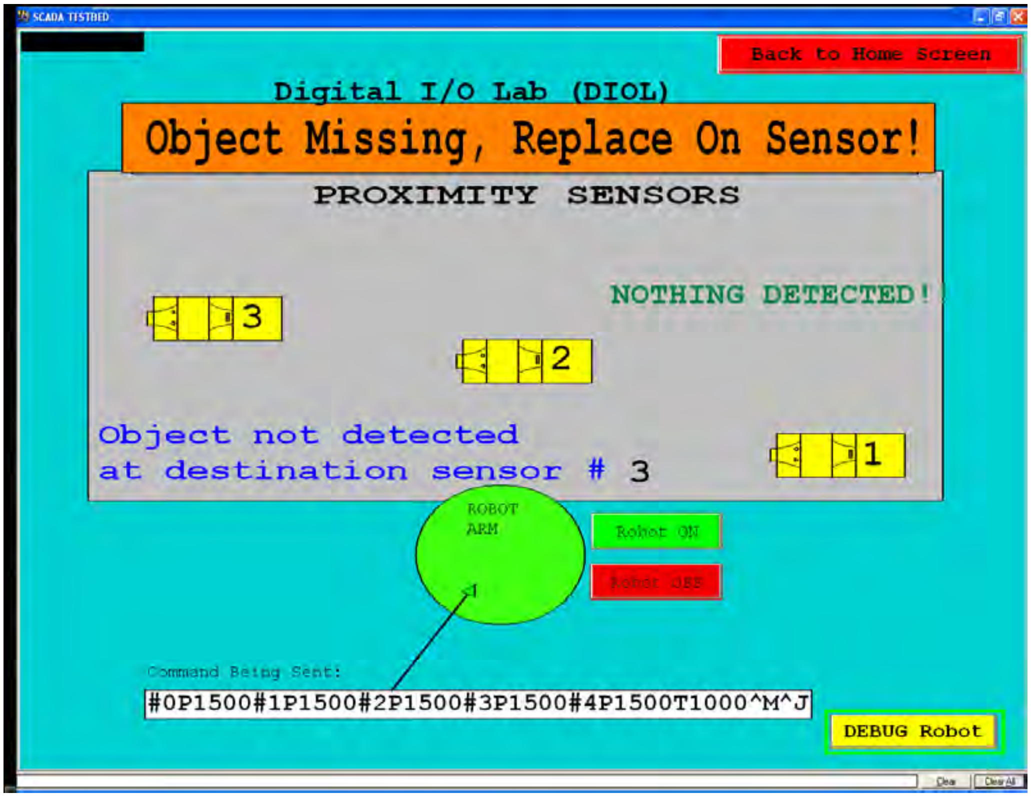


Step 5. HMI: click the green “1 to 3 move” button

Observation: The robot arm moves to pick up the barrel object and moves the object to sensor #3. (the next step is done immediately after the completion of this observation.)

Step 6. Proximity Sensors: AFTER the robot arm delivers the barrel object to sensor #3 and BEFORE the robot arm completes its cycle: remove the barrel from sensor #3. Symbolizing the barrel object was not delivered as expected.

Observation: HMI: an error is displayed indicating that the object is missing and the location the object should be replaced.



Digital I/O Lab (DIOL)

Object Missing, Replace On Sensor!

PROXIMITY SENSORS

NOTHING DETECTED!

Object not detected at destination sensor # 3

Command Being Sent:

```
#0P1500#1P1500#2P1500#3P1500#4P1500T1000^M^J
```

Step 7. Replace the object on sensor #3

Observation: HMI: sensor icon #3 is green. The green “3 to 2 move” and “3 to 1 move” buttons are visible

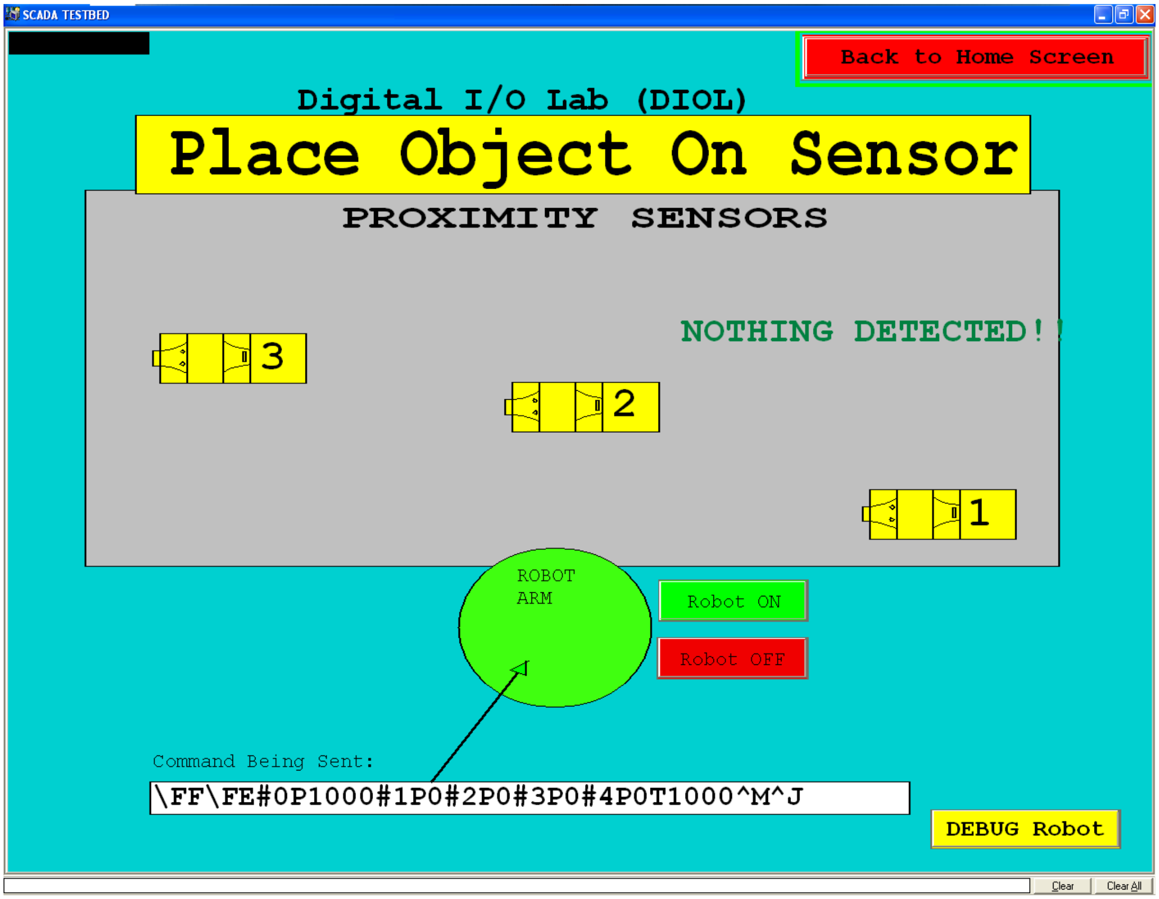
Observation: Proximity Sensor: both the green and yellow LED indicators located on the sensor #3 are illuminated

Step 8. HMI: click on the red “Robot OFF” button

Observation: The robot arm falls to the “off” position and remains there

Observation: HMI: the center “ROBOT ARM” graphic eventually turns green. Signifying the robot is ready for another command.

Step 9. Remove the barrel object located on sensor #3



Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Step 10. HMI: click the red “Back to Home Screen” button

Observation: The lab has been returned to the Ready State

Step 11. The test TE3 is complete

TEST TE4: Robot arm disturbs wrong object

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the blue “DIGITAL IO LAB” button.

Step 3. Place the barrel object on proximity sensor # 1.

Observation: HMI: sensor icon #1 is green. The green “1 to 2 move” and “1 to 3 move” buttons are visible

Observation: Proximity Sensor: both the green and yellow LED indicators located on the sensor #1 are illuminated

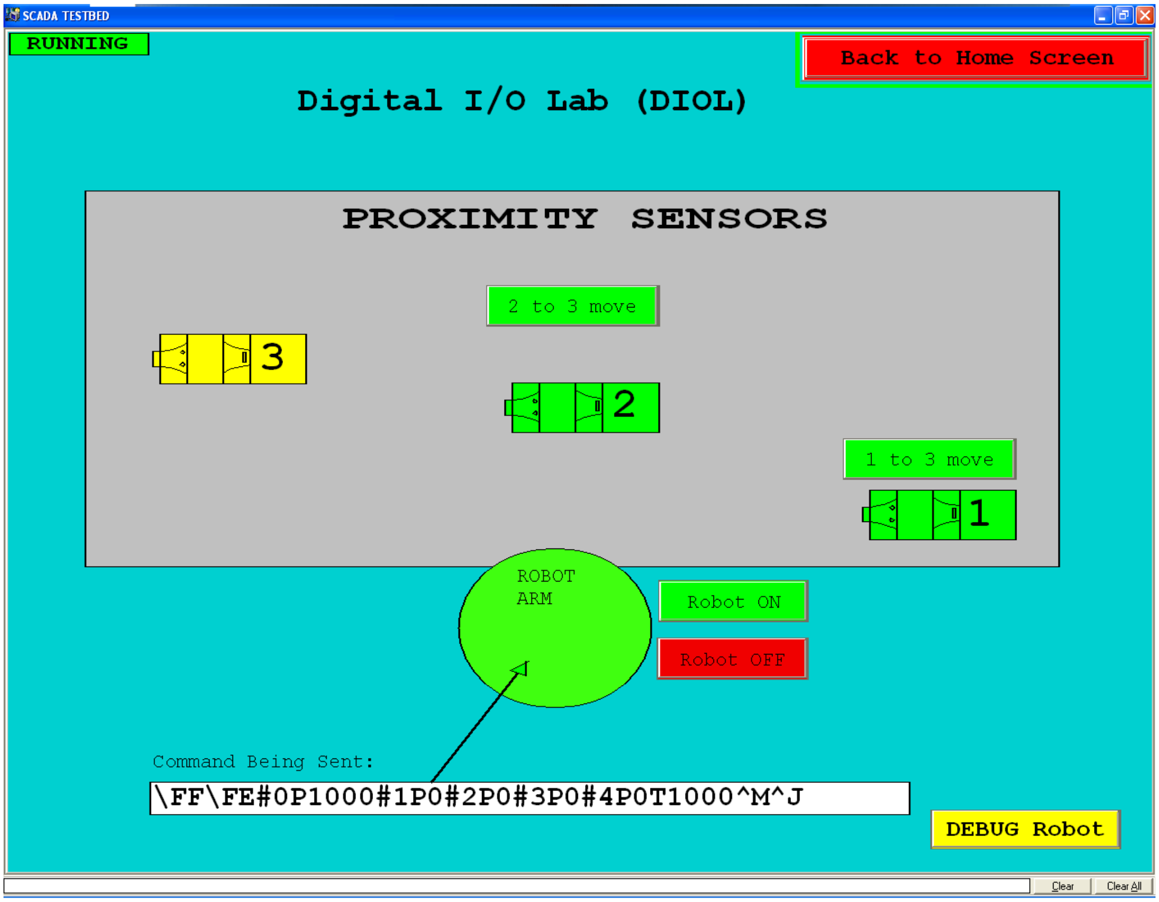
Step 4. Place an additional object on proximity sensor # 2.

Observation: HMI: sensor icon #1 and #2 are green. The green “1 to 3 move” and “2 to 3 move” buttons are visible

Observation: Proximity Sensor: both the green and yellow LED indicators located on the sensor #1 and #2 are illuminated

Step 5. HMI: click the green “Robot On” button

Observation: Robot arm moves to the “stow” position.



Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Step 5. HMI: click on the green “1 to 3 move” button

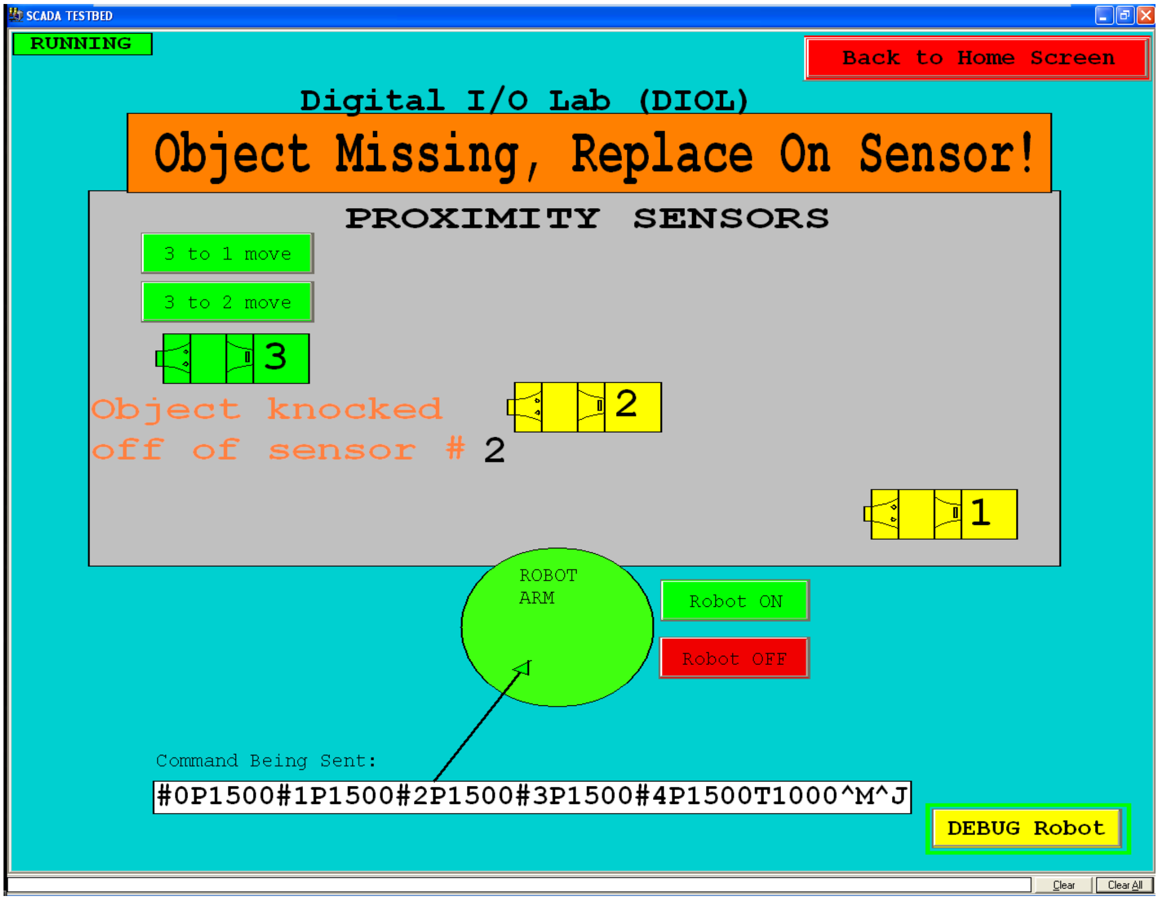
Observation: Robot arm proceeds to move the object located at sensor #1 to sensor #3. (proceed to next step during the movment)

Step 6. Proximity Sensors: before the robot arm completes the movement from sensor #1 to #3, remove the object located at sensor #2.

Observation: Ensure the robot arm successfully delivers the object to sensor #3

Error Occured: If the robot arm is unsuccessful in delivering the object to sensor #3, repeat the test.

Observation: HMI: an error is displayed indicating an object is missing and was knocked off of sensor #2.



Object Missing, Replace On Sensor!

Object knocked off of sensor # 2

Command Being Sent:

```
#0P1500#1P1500#2P1500#3P1500#4P1500T1000^M^J
```

Step 7. Proximity Sensor: return the object to sensor #2

Observation: HMI: all errors clear

Observation: HMI: sensor icon #2 and #3 are green. The green “2 to 1 move” and “3 to 1 move” buttons are visible

Observation: Proximity Sensor: both the green and yellow LED indicators located on the sensor #2 and #3 are illuminated

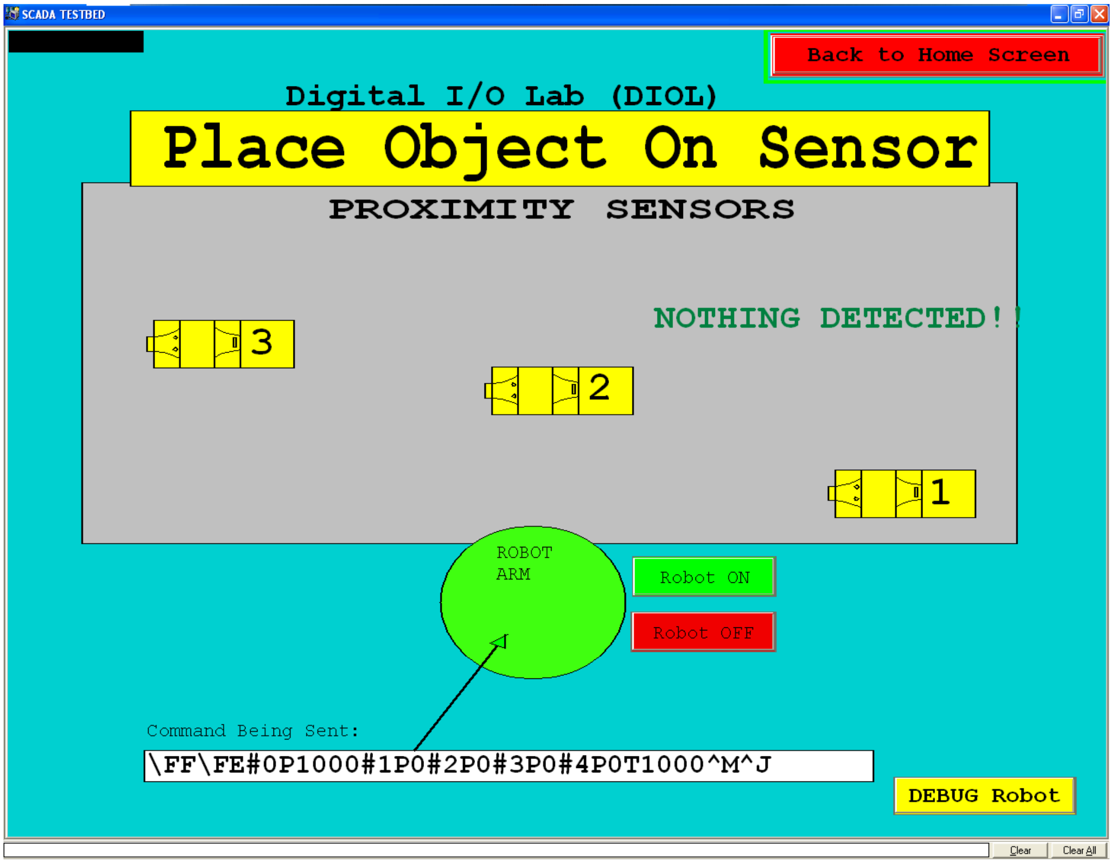
Step 8. Proximity Sensor: remove the objects from sensors #2 and #3

Observation: HMI: an error is displayed indicating that none of the sensors are active. The sensors are all yellow.

Step 9. HMI: click on the red “Robot OFF” button

Observation: The robot arm falls to the “off” position and remains there

Observation: HMI: the center "ROBOT ARM" graphic eventually turns green. Signifying the robot is ready for another command.



Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Step 10. HMI: click the red "Back to Home Screen" button

Observation: The lab has been returned to the Ready State

Step 11. The test TE4 is complete

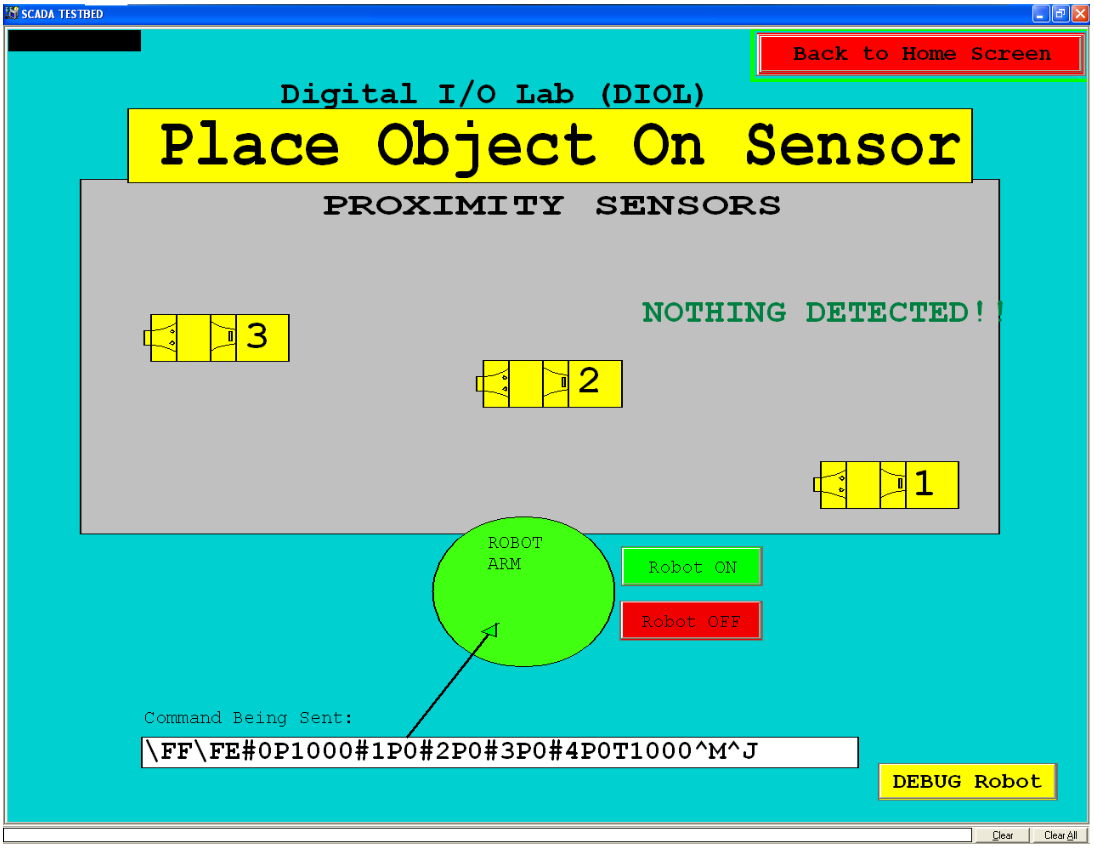
TEST TE5: No proximity sensor active

Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the blue "DIGITAL IO LAB" button.

Observation: HMI: Error is displayed indicating that none of the sensors are active and to place an object on the sensor.

Observation: Proximity Sensors: No objects exist on any of the 3 proximity sensors. All Proximity sensors have only a green LED lit.



Digital I/O Lab (DIOL)

Place Object On Sensor

NOTHING DETECTED!!

Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Step 3. HMI: click the red "Back to Home Screen" button

Observation: The lab has been returned to the Ready State

Step 4. The test TE5 is complete

TEST TE6: Unable to move objects with robot arm, no open sensors exist

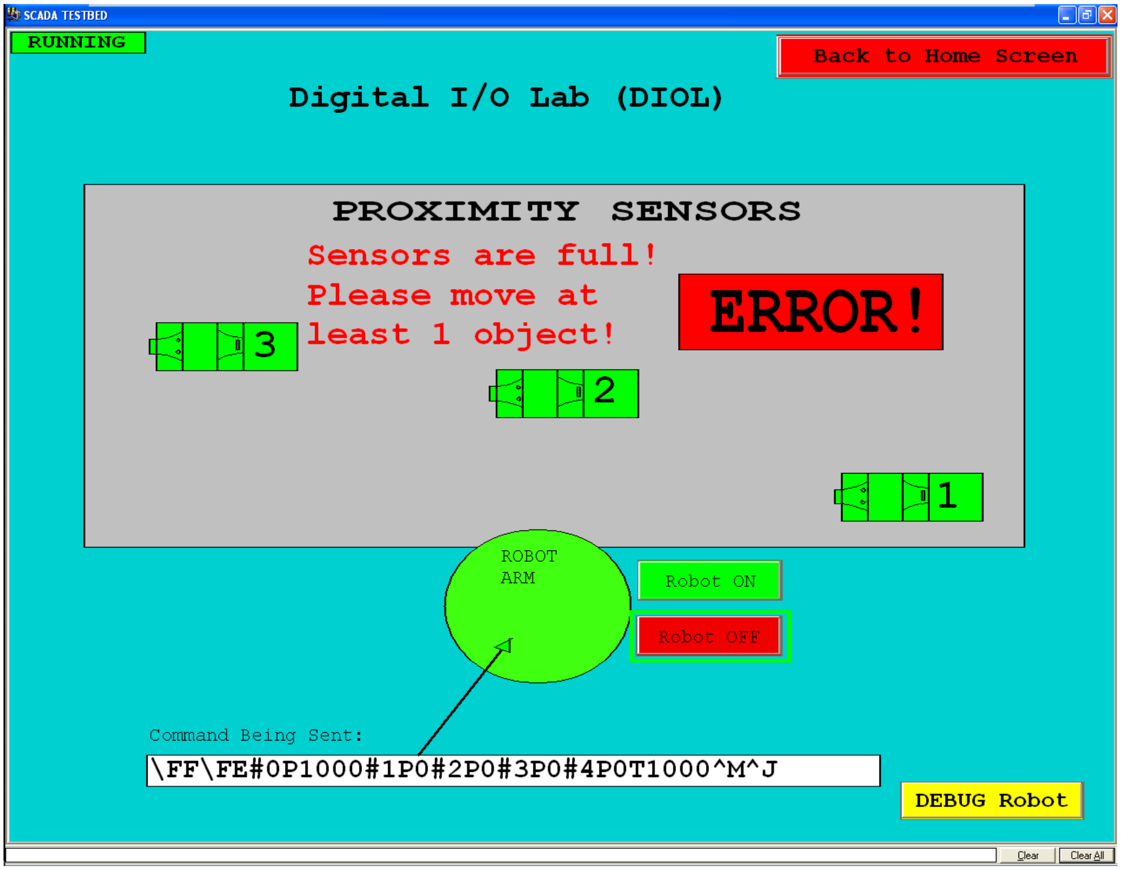
Step 1. Place the lab in the Ready State as outlined in chapters F1-F3.

Step 2. HMI: click on the blue “DIGITAL IO LAB” button.

Step 3. Proximity sensors: place an object on sensors #1, #2 and #3

Observation: HMI: No “move” buttons exist as an option to move the robot arm. The arm cannot perform a movement of objects without an open sensor.

Observation: HMI: sensor icon #1, #2 and #3 are green. An error is displayed indicating that all the sensors are full and at least 1 object must be removed from a sensor.



Sensors are full!
Please move at
least 1 object!

ERROR!

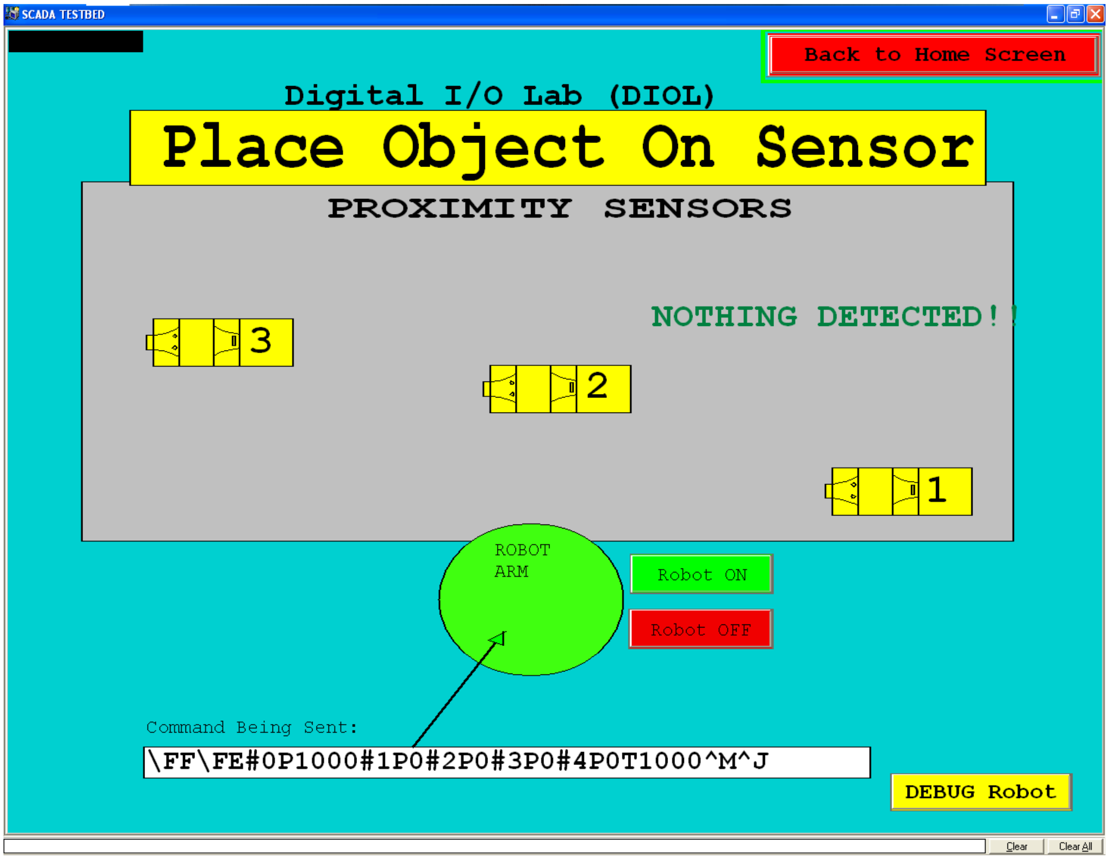
Command Being Sent:

```
\FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J
```

Observation: Proximity Sensor: both the green and yellow LED indicators located on the sensor #1, #2 and #3 are illuminated

Step 4. Proximity Sensor: remove all objects from sensors #1, #2, and #3

Observation: HMI: the display returns to the Steady state screen.



Command Being Sent: \FF\FE#0P1000#1P0#2P0#3P0#4P0T1000^M^J

Step 5. HMI: click the red “Back to Home Screen” button

Observation: The lab has been returned to the Ready State

Step 6. The test TE6 is complete

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Department of Homeland Security: National Cybersecurity and Communications Integration Center. (2013, Dec.). “ICS-CERT year in review.” [Online]. https://ics-cert.us-cert.gov/sites/default/files/documents/Year_In_Review_FY2013_Final.pdf

[2] M. Mimoso. (2014, Jul. 7). Motives behind Havex ICS malware campaign remain a mystery. [Online]. Available: http://threatpost.com/motives-behind-havex-ics-malware-campaign-remain-a-mystery

[3] K. Wilhoit. (2014, Jul. 17). Havex its down with OPC. [Online]. http://www.fireeye.com/blog/technical/targeted-attack/2014/07/havex-its-down-with-opc.html

[4] S. Curtis. (2014, Feb. 11). ‘The mask’ cyber spying operation targets government agencies. [Online]. http://www.telegraph.co.uk/technology/internet-security/10630272/The-Mask-cyber-spying-operation-targets-government-agencies.html

[5] J. Wagstaff. (2014, Apr. 23). All at sea: global shipping fleet exposed to hacking threat. [Online]. Available: http://www.reuters.com/article/2014/04/24/us-cybersecurity-shipping-idUSBREA3M20820140424

[6] D. Antanitus. (2014, Apr.). Sailor-less ships soon could be a reality in U.S. Navy. [Online]. Available: http://www.nationaldefensemagazine.org/archive/2014/April/Pages/Sailor-LessShipsSoonCouldBeaRealityinUSNavy.aspx

[7] The Department of Homeland Security. (2013, Jun.). “Executive order 13636: improving critical infrastructure cybersecurity.” [Online]. http://www.dhs.gov/sites/default/files/publications/dhs-eo13636-analytic-report-cybersecurity-incentives-study.pdf

[8] White House. (2013, Feb.) “Presidential policy directive—Critical intrastructure security and resilience.” [Online]. http://www.whitehouse.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil

[9] Annarita Giani et al., “A testbed for secure and robust SCADA systems,” *IEEE Real-Time and Embedded Technology and Applications Symposium*, St. Louis, MO, 2008, p. 4.

[10] T. Morris, R. Vaughn, and D. Yoginder, "A testbed for SCADA control system cybersecurity reasearch and pedagogy," in *7th Cyber Security and Information Intelligence Research Workshop*, Oak Ridge, TN, 2011, p. 4.

[11] I. N. Fovino, M. Masera, L. Guidi, and G. Capri, "An exprimental platform for assessing SCADA vulnerabilities and coutermeasures in power plants," in *Human System Interactions (HSI) 2010*, Rzeszow, 2010, pp. 679–686.

[12] R. G. Bensing, "An assesment of vulerabilities for ship-based control systems," M.S. thesis, Dept. Computer Science, Naval Postgraduate School, Monterey, CA, 2009.

[13] United States General Accounting Office. (2005, Jun.). Critical infrastrucutre protection: challenges in securing control systems. [Online]. Available: http://www.gao.gov/new.items/d04140t.pdf

[14] T. Scherer and J. Cohen, "The evolution of machinery control systems support at the naval ship systems engineering station," *Naval Engineers Journal*, vol. 123, no. 2, pp. 85–109, Jun. 2011.

[15] M. P. Ward, "An architectural framework for describing supervisory control and data acquisition (SCADA) systems," M.S. thesis, Dept. Computer Science, Naval Postgraduate School, Monterey, CA, 2004.

[16] Rockwell Automation. (2004, Jun.). *SLC 500 4-Channel Analog I/O Modules*. [Online]. Available: http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1746-um005_-en-p.pdf

[17] Rockwell Automation. (2000, Feb.). *SLC 500 Analog Input Module* (Catalog number 1746-NI8). [Online]. Available: http://literature.rockwellautomation.com/idc/groups/literature/documents/in/1746-in006_-en-p.pdf

[18] Rockwell Automation. (2004, Jun.). *SLC 500 Modular Hardware Style*. [Online]. Available: http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1747-um011_-en-p.pdf

[19] Rockwell Automation. (2011, Jul.). *RSView32 Users Guide*. [Online]. Available: http://literature.rockwellautomation.com/idc/groups/literature/documents/um/vw32-um001_-en-e.pdf

[20] *Rockwell Software RSLogix500 SLC-500 Programming*. (2011, Jun.). Dogwood Valley Press, LLC., Rolla, MO. [Online]. Available: http://dogwoodvalleypress.com/uploads/moreresources/07272011094146.pdf

[21] IEEE Criteria for Programming Industrial Automation Systems, *IEC 61131-3: Programming Industrial Automation Systems:Concepts and Programming Languages, Requirements for Programming Systems, Decision-making Aids*, 2nd ed. Berlin, Germany: Heidelberg:Springer-Verlag, 2010.

[22] Rockwell Automation. (2008, Nov.). *SLC 500 Instruction Set*. [Online]. Available: http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1747-rm001_-en-p.pdf

[23] Lynxmotion. (2010, Jun.). *SSC-32 Manual*. [Online]. Available: http://www.lynxmotion.com/images/html/build136.htm

[24] Lynxmotion. (2011, Apr.). *SSC-32 Binary Commands*. [Online]. Available: http://www.lynxmotion.com/images/html/build177.htm

[25] TURCK Sensors USA. (2014, Aug.). *TURCK Level Sensors—Capacitive*. [Online]. Available: http://old.turck.us/illustrations/B1008_G17.pdf

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California